

Operating Instructions

Personal Computer
JR-200U



Panasonic

Before operating this set, please read these instructions completely.

Operating Instructions

U005-91



Copyright © 1985

TABLE OF CONTENTS

| | page |
|---|------|
| 1. Introduction | 3 |
| 2. Important Don'ts | 4 |
| 3. Parts Name | 5 |
| 4. Connection to Cassette Tape Recorder | 7 |
| 5. Operations | 8 |
| 1. Starting | 8 |
| 2. Keyboard Operations | 10 |
| (1) Normal Mode (Alphanumeric Character Mode) | 11 |
| (2) Graphic Mode | 11 |
| (3) Function Mode (Control Mode) | 11 |
| (4) BREAK Key | 12 |
| (5) Screen Editor | 13 |
| 6. Display Screen | 15 |
| 1. Color Selection | 15 |
| 2. Frame Display | 16 |
| 7. Operation of Cassette Tape Recorder | 17 |
| 1. Connection | 17 |
| 2. Baud Rate Setting | 17 |
| 3. Program Recording (SAVE) | 18 |
| 4. Program Loading (LOAD) | 20 |
| 5. Recording Check (VERIFY) | 21 |
| 6. Machine Language Recording and Loading (MSAVE and MLOAD) | 22 |
| 7. Merge of Two Programs (MERGE) | 22 |
| 8. Other Operations | 23 |
| 8. Specifications of BASIC Language | 25 |
| 1. Program Format | 25 |
| 2. Variables | 25 |
| 3. Constants | 26 |
| 4. Expressions and Operations | 27 |
| 5. Array Declaration and Element Reference | 29 |
| 6. BASIC Commands and Statements (Sentences) | 31 |
| 7. BASIC Functions | 39 |
| 8. Command of the Monitor | 43 |
| 9. Error Messages | 44 |

| | |
|---------------------------|----|
| 9. Specifications | 47 |
| 10. Troubleshooting | 49 |

Appendix

| | |
|--|----|
| 1. Memory Map | 51 |
| 2. Color Code Table | 51 |
| 3. Attribute Area Data Allocation Table | 52 |
| 4. Color Data Allocation Table of Semigraphic Pattern | 52 |
| 5. Variables of PLOT Command and its Screen Correspondence | 52 |
| 6. User's Definition Character Function | 53 |
| 7. Use of Character Generator | 56 |
| 8. Display Code Table | 57 |
| 9. Character Code Table | 57 |
| 10. Musical Performance Program | 58 |

1. INTRODUCTION

Thank you for purchasing the Panasonic personal computer JR-200U.

The Panasonic personal computer JR-200U is designed to satisfy needs of a beginner as well as an experienced user.

Panasonic personal computer JR-200U features:

- (1) Color indication
- (2) Operation of floating point arithmetic expressions
- (3) Use of data files
- (4) Performance of triad chords related to the display
- (5) Conveniently used monitor for machine language
- (6) High speed recording (with a tape recorder) of 2400 baud

This user manual describes fundamental ways to use the JR-200U.

2. IMPORTANT DON'TS

Use the JR-200U in accordance with following notes. Any improper operation may cause trouble.

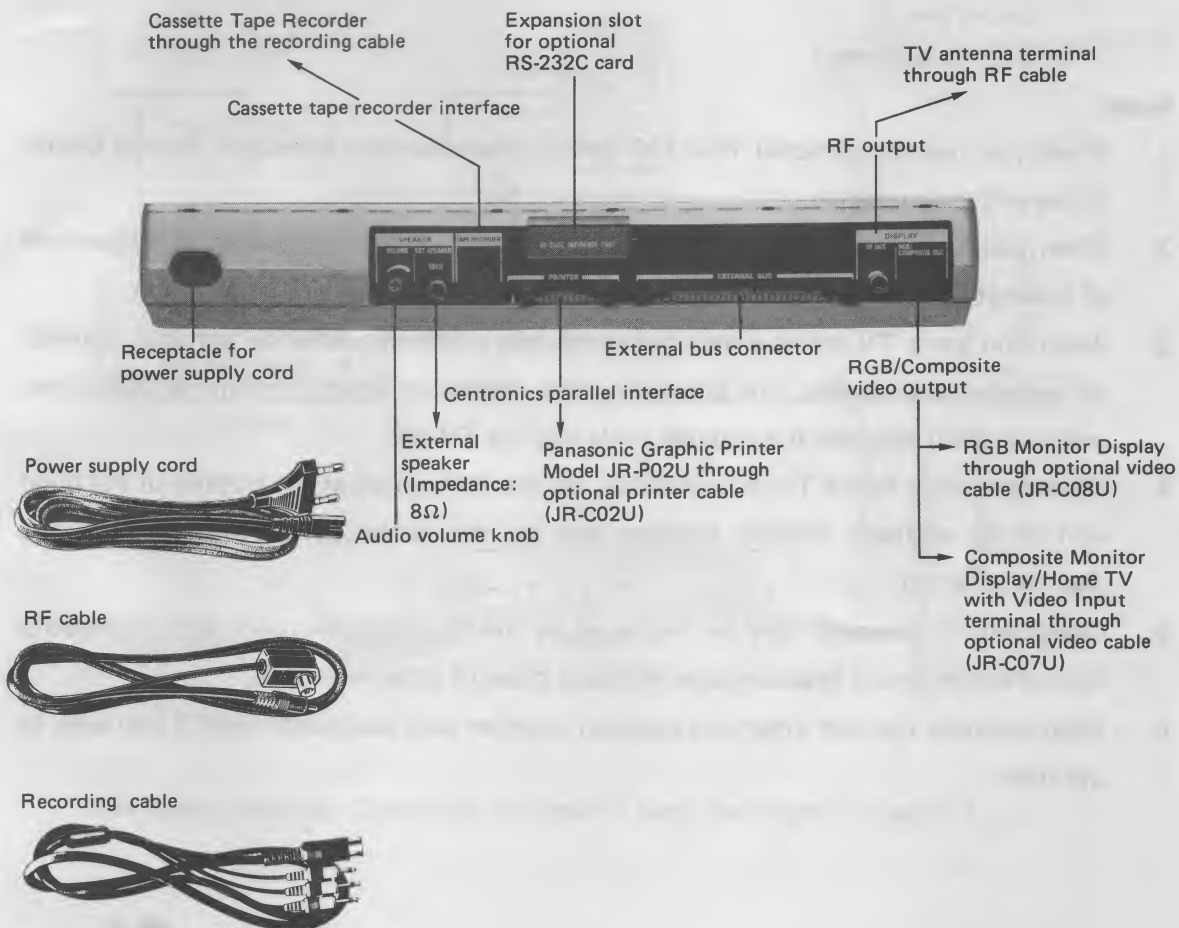
- Place the main unit of JR-200U away from places of high or low temperature, direct sunshine (the main unit may be discolored), or dusty places. Locate it away from a place where the change of temperature is abrupt.
- Do not spill coffee, juice, tea, etc. over the main unit.
- Do not disassemble the main unit. When trouble is found, contact the nearest Panasonic Service Center immediately.
- Do not use volatile solvents such as paint thinner or benzene for cleaning. Wipe the surface of the main unit with a dry cloth.
- Do not connect the RF converter output terminal of the JR-200U to a home TV master antenna terminal.
- Do not drop or throw the main unit. Strong physical shocks must be avoided.
- Locate the main unit away from a receiver such as a radio receiver since noise may be produced in the radio speaker when JR-200U is operated nearby.
- Set the channel selection switch on the under side of the main unit to 3 or 4 which does not correspond to any local TV station. Do not connect any device to the RF output terminal except the dedicated RF cable. Do not leave the power supply switch ON when the microcomputer is not in use.
- Do not use any equipment other than optional units, to expand your system.
- Do not connect a home TV antenna, etc. to the RF converter output terminal of the JR-200U.

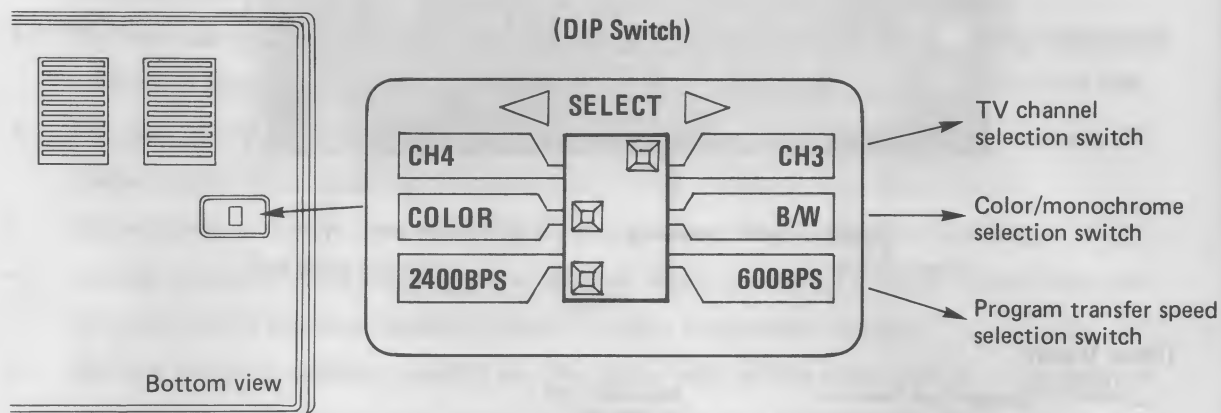
3. PARTS NAME

(Top View)



(Rear View)





Notes:

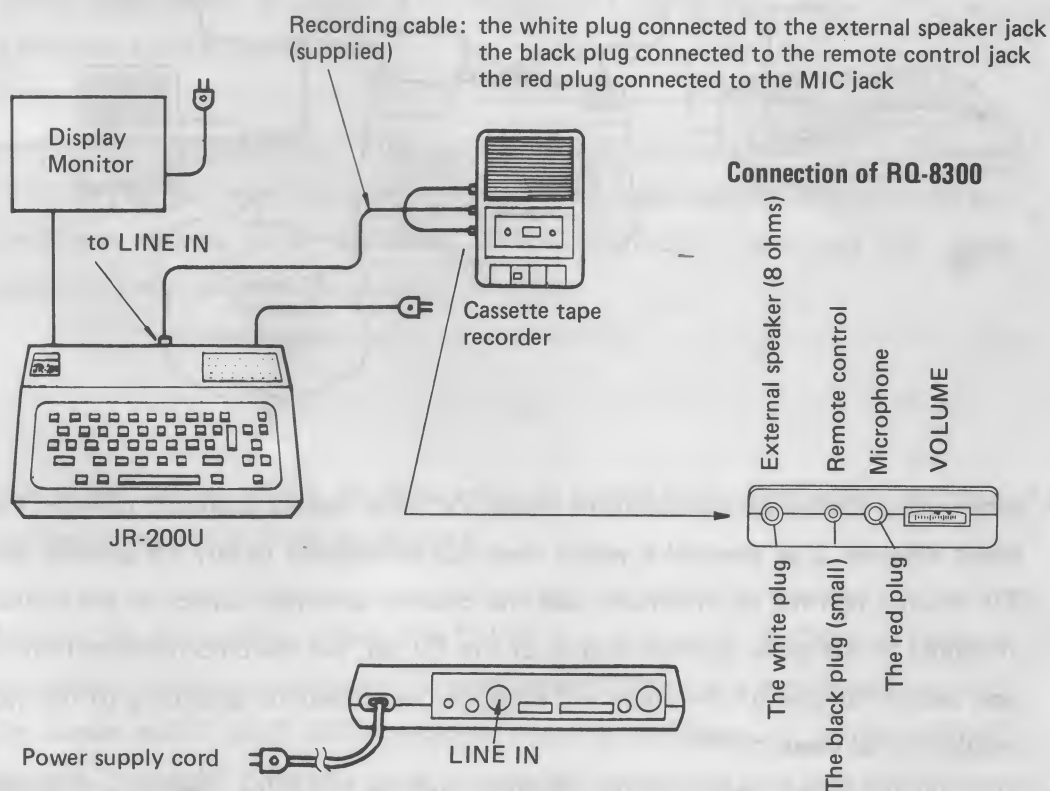
1. When you use the optional RS-232C board, please contact Panasonic Service Center in the written guarantee.
2. When you use a monochrome display as a monitor, set the DIP switch at the bottom of the unit to the monochrome position to obtain clearer images on the screen.
3. When you use a TV set as a monitor which has a 300-ohm antenna terminal, connect an optional transformer (for converting the impedance from 75 ohms to 300 ohms, and vice versa) between the antenna cable and the TV set.
4. When you use a home TV as a monitor, set the DIP switch at the bottom of the main unit to an available channel position and set the home TV to the corresponding channel position.
5. Characters or diagrams may be displayed in the monochrome mode and misregistration of colors occurs in accordance with the types of color monitors.
6. Read carefully the user's manuals supplied together with peripheral units if you wish to use them.

4. CONNECTION TO CASSETTE TAPE RECORDER

Cassette tape recorder RQ-8300, specially designed for JR-200U, is recommended. Using the supplied recording cable, connect the red plug to the MIC jack, the white plug to the external speaker jack or monitor and the black plug to the remote control jack. Leave the black plug unplugged when you use a tape recorder having no remote control jack.

If you use a tape recorder which is not specially designed for JR-200U, remove the black plug from the remote control jack in the rewinding and fast forwarding modes. (The black plug need not be removed from the remote control jack when the tape recorder designed for JR-200U is used.)

RQ-8300 enables high-speed transfer from the cassette tape at 2400 baud for both recording and playing. The other cassette tape recorders must be used at 600 baud.

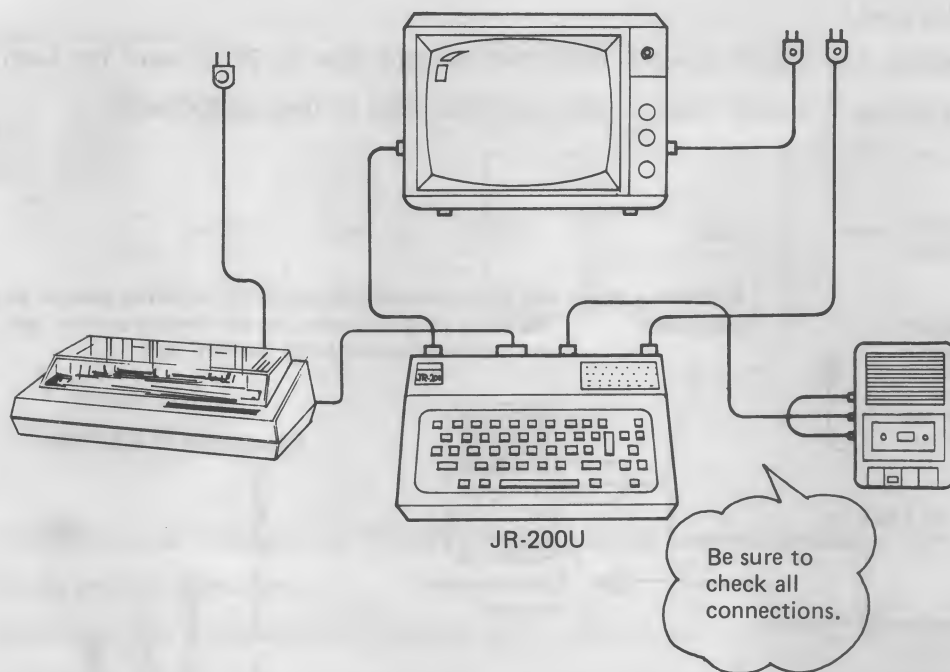


For details, refer to "Operation of Cassette Tape Recorder" on page 17.

5. OPERATIONS

1. Starting

- (1) All preparations and checking must be made before turning ON the power supply of the JR-200U. Check if power supplies of all devices (such as a monitor, a printer, cassette tape recorder and so on) connected to JR-200U are turned OFF. If some devices are turned ON, turn them OFF.



- (2) When using a color or monochrome home TV for a display monitor, turn on TV and select channel 3 or channel 4 which does not correspond to any TV station. Set the TV volume control to minimum. Set the channel selection switch at the bottom of JR-200U to the same channel as that of the TV set. Set the color/monochrome selection switch to one of the color and monochrome positions according to the type of monitor to be used.
- (3) Turn on the power switch of the JR-200U and the following message will appear on the display.
- (4) Set the volume control on the back of JR-200U to an adequate level for producing key-in sounds.
- (5) Turn on power switches of peripheral devices connected to the JR-200U.

```
J R   B A S I C   5 . 0  
( C )   1 9 8 2   b y  
M a t s u s h i t a   S y s t e m   E n g i n e e r i n g  
F r e e   B y t e s   3 0 7 1 6  
R e a d y  
■
```

If this message does not appear, check connections of the devices again.

- (6) The message described in item (3) indicates that JR-200U is ready to receive instructions from you. "Free Bytes 30716" indicates an available BASIC Program area of 30716 bytes. Refer to chapter 8 "Operation of Cassette Tape Recorder" so as to load a program from a cassette tape.

Notes: Do not turn ON/OFF power switches of devices connected to the main unit. Do not connect/disconnect the connecting cords between the main unit and peripheral devices while the power switch of JR-200U is turned ON. These operations may cause malfunction of JR-200U.

- (7) The power switch of the JR-200U must be turned OFF after power switches of all devices connected to it are turned OFF.

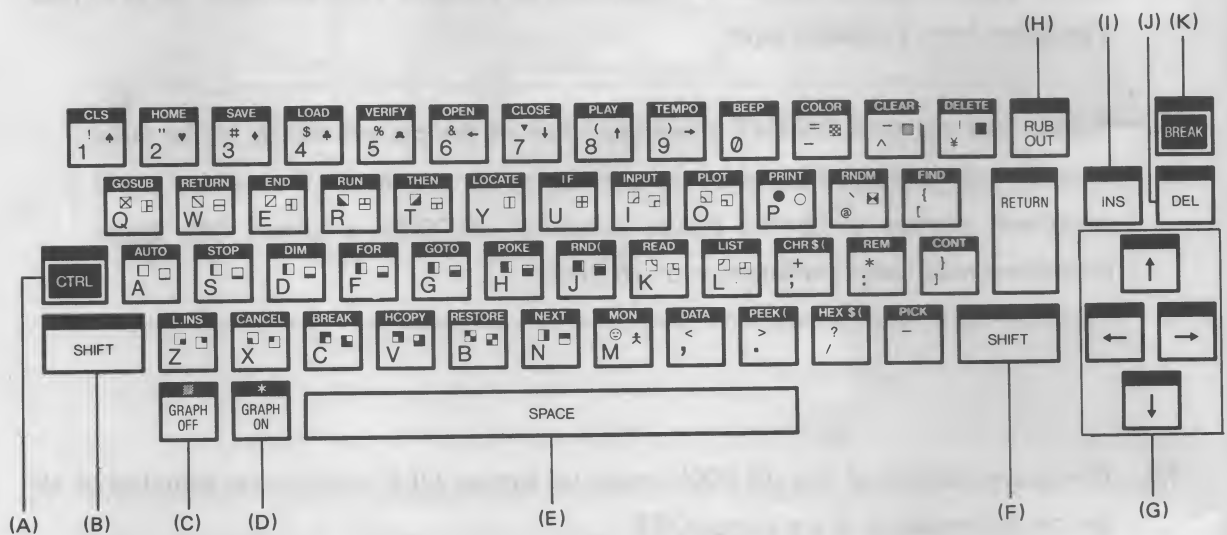
Notes: Do not quickly repeat the ON/OFF operation of JR-200U or connected devices. It may cause trouble or malfunction. When you erroneously turn off the power switch of JR-200U programs stored in the main unit of the JR-200U are lost. Important programs must be recorded and stored on a tape before turning the computer off.

2. Keyboard Operations

JR-200U has three key input modes consisting of the normal mode, the graphic mode, and the function mode. These modes can be used at 5 separate stages together with the **SHIFT** key. This enables high-speed program creation with the keys, a decrease in erroneous typing, and formation of graphic figures of various types by using the large variety of graphic symbols.

When you turn ON the power switch or perform the break operation (Hard Break), the keyboard is immediately initiated in the normal mode automatically.

Keyboard Arrangement



(1) Normal Mode (Alphanumeric Character Mode)

Press the **GRAPH OFF** key (c) to change the keyboard from the graphic mode to the normal mode. In the normal mode, the cursor on the screen is shown by "■". Press a key to input an alphanumeric character marked (in white) on the key top. Simultaneously press the **SHIFT** key (B or F) and a character key to input a capital letter. Simultaneously press the **SHIFT** key and a key other than a character key to input the symbol marked (in white) on the upper left of the key top.

(2) Graphic Mode

Press the **GRAPH ON** key (D) to change the keyboard mode to the graphic mode. The cursor on the screen is shown by "*". In the graphic mode, you can input a graphic symbol (shown in blue) on the key top. Press the key only to input the symbol on the right side of the key top.

Simultaneously press the **SHIFT** key and the key to input the symbol on the left side of the key top.

(3) Function Mode (Control Mode)

Simultaneously press the **CTRL** key (A) and a key in the normal mode to input a key word indicated in the upper frame of the key top. For example, simultaneously press the **CTRL** key and the **P** key to display a key word "PRINT _" on the display. A space of one character is automatically added after the key word (shown by _ symbol in this manual). However, there are some exceptions that the key word is not displayed on the display but functions as a command. The exceptional keys are as follows:

| Key name | Command | Function |
|----------|-------------------------|---|
| 1 | CLS | To clear the display. |
| 2 | HOME | To move the cursor to the upper left corner of the display. |
| Z | L. INS (LINE INSERT) | To move the line with the cursor upward by one line and to make one blank line. |
| X | CANCEL | To cancel an input. |
| C | BREAK | To stop the program operation. |
| V | HCOPY | To print out the content currently displayed on the screen to the printer. |

Note: Press the **SPACE** key to insert a space in any mode.

(4) **BREAK** key

Press the **BREAK** key to immediately interrupt all operations and set the computer in the Ready mode.

Differences of functions between the **BREAK** (CTRL + C) in the function mode and the **BREAK** key located on the upper right of the keyboard are as follows:

| Name | Software break | Hardware break |
|-------------|---|---|
| Operation | BREAK (CTRL + C) by function keys | BREAK key located at the upper right corner of the keyboard |
| Explanation | <ol style="list-style-type: none">1) Use the Soft Break to interrupt a program which must be restarted by the CONT command.2) The program is interrupted after the end of operation currently performed, and the computer is then set in the Ready mode.3) The program can be continued by the CONT command.4) Trouble caused by machine language cannot always be cancelled.5) Music performance is not halted.6) A cassette does not stop during the "LOAD" operation for placing a program into the JR-200U, or during the "SAVE" operation for recording a program on magnetic tape.7) The program is retained in the JR-200U and is capable of restarting the execution. | <ol style="list-style-type: none">1) Use the Hard Break to interrupt a program which need not to be restarted by the CONT command or when an uncontrolled operation occurs.2) All operations are halted and the computer is set in the Ready mode.3) The program cannot be continued by the CONT command.4) Trouble caused by machine language can be cancelled.5) Music performance is halted.6) A cassette tape is stopped during the "LOAD" operation or during the "SAVE" operation.7) The program is retained in the JR-200U and is capable of restarting the execution. |

NOTES: If a program is not interrupted by either one of above break commands, turn the power supply switch OFF and ON again.

When an interruption of the program by a break command occurs, "Break in line n" (n indicates the line number) is displayed to indicate the number of the line where the program is interrupted.

(5) Screen Editor

An error may occur when loading or running a program. Statements and programs displayed containing an error can be corrected on the screen. The functions described below may be utilized for this purpose.

| Key | Function |
|--|---|
| ← ↓ ↑ → (G) | To move the cursor one space in the direction shown by arrow. |
| RUB OUT (H) | To delete the character immediately before the current cursor position and move the cursor to the position immediately before the current position. |
| DEL (J) | To delete the character above the cursor and carry the following lines. |
| L.INS (CTRL + Z) | To insert a blank line. |
| CANCEL (CTRL + X) | To cancel an input and move the cursor to the beginning of the line. |
| INS (I) | To change the normal mode to the insert mode for inserting a character. Press this key again to restore the normal code. |
| HOME (CTRL + 2) | To move the cursor to the initial position of the uppermost line. |
| CLS (CTRL + 1) | To clear the display and move the cursor to the initial position of the uppermost line. |
| RETURN | To input symbols or characters in the current line marked with the display cursor. |

← , ↑ , ↓ , and → are used to move the cursor in a desired direction to correct the sentences. Correction on the screen is performed by utilizing the **DEL** , **INS** , and **RUBOUT** keys. When the **RETURN** key is pressed after correction on the screen, the corresponding program within the memory is corrected in the same manner as displayed on the screen. The **RETURN** key may be pressed at any time regardless of the cursor position on a given line.

For deleting an arbitrary line, input the line number and press the **RETURN** key. For deleting a series of lines, utilize the **DELETE** command.

For inserting a desired sentence between lines, input a line before the specified line; the line number will be arranged in order again.

During the period between completion of the **STOP** sentence (command) and execution of the **CONT** command, a direct command may be executed, but program correction cannot be performed.

When a long program is reviewed on the screen by executing the **LIST** command, the images on the screen are scrolled up. A stationary image may be obtained by pressing a key other than the **BREAK** key, the **CTRL** key or the **SHIFT** key. Press the key again to review the program step by step.

Use the **FIND** command to list the same type of sentences. In the **FIND** command, a line containing a specified character-string is searched for and the entire statement is displayed for convenience of correcting errors. A stationary image may be obtained by pressing a key.

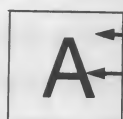
6. DISPLAY SCREEN

JR-200U creates 8 colors on the screen.

Follow the procedure described below.

1. Color Selection

Specify colors of numerals, letters, and the background by color codes using COLOR sentences.



| | | |
|-------------|----------|-------------------|
| Color code: | 0 black | 4 green |
| | 1 blue | 5 sky blue (cyan) |
| | 2 red | 6 yellow |
| | 3 purple | 7 white |

Format: COLOR \square [\langle Expression 1 \rangle] [, [\langle Expression 2 \rangle] [, [\langle Expression 3 \rangle]]

Expression 1 . . . To specify character and graphic colors.

Expression 2 . . . To specify a background color.

| Value of Expression (Color Code) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------------------------------------|-------|------|-----|--------|-------|-------------|--------|-------|
| Color | black | blue | red | purple | green | sky blue | yellow | white |

Expression 3 . . . To select the display mode.

| Value | Mode | Function |
|-------|------------------------------|---|
| 0 | Normal mode | To release the user's definition mode. |
| 1 | User's definition mode | To display a user's definition character. (Refer to Appendix on page 53) |
| 2 | Inversion mode | To invert character and background colors into specified colors, respectively. |
| 3 | Background color change mode | To change a background color to a color specified by sentence 2. (The color of a one-character region specified by a PLOT statement cannot be changed.) |

Reference Program:

Let's try the following program.

(The display color is changed.)

```
10 FOR K=0 TO 7
20 COLOR 7,0
30 PRINT "BACKGROUND";K
40 FOR I=0 TO 7
50 COLOR I,K
60 PRINT "COLOR___";
70 NEXT I
80 NEXT K
90 COLOR 7,0
```

2. Frame Display

- A color frame can be formed: each of the upper and lower sides of the frame corresponds to two lines, and each of the right and left sides corresponds to four digits or characters. Enter the following POKE sentence.

POKE _ \$CA00, n (where n is the color code)

For example, if you wish a blue frame, enter:

POKE _ \$CA00, 1

Reference Program:

Execute the following program to create frames each having one of 8 colors.

```
10 CLS ← Press the C, L, S keys.
20 FOR I=0 TO 7
30 POKE $CA00, I
40 LOCATE 10, 11
50 PRINT "POKE $CA00, "; I
60 FOR K=1 TO 1500 : NEXT K
70 NEXT I
80 GOTO 10
```

7. OPERATION OF CASSETTE TAPE RECORDER

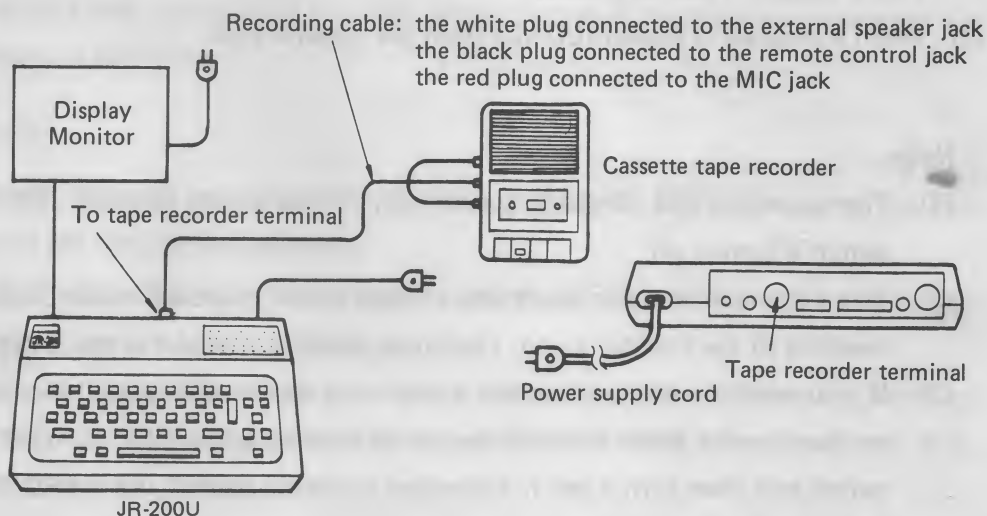
The JR-200U has functions to fetch a program recorded on a cassette tape and to store a program on the cassette tape.

Use the cassette tape recorder in accordance with the procedure that follows.

1. Connection

Using the supplied recording cable, connect the red plug to the MIC jack, the white plug to the external speaker jack and the black plug to the remote control jack. Leave the black plug unplugged when you use a tape recorder having no remote control jack. If you use a tape recorder which is not specifically designed for JR-200U, remove the black plug from the remote control jack in the rewinding and fast forwarding modes. (The black plug need not be removed from the remote control jack when the tape recorder RQ-8300 designed for JR-200U is used.)

When you use the tape recorder (model RQ-8300), you can save/load the program at a transfer speed of 2400 baud. If you use any tape recorder other than model RQ-8300, set the transfer speed to be 600 baud with the switch on the bottom of the JR-200U.



2. Baud Rate Setting

Set the transfer speed selection switch to the 2400- or 600-baud position on the bottom panel of the JR-200U before power is supplied. If so set, the internal program is preset for 2400 or 600 baud when you turn on the power switch.

The transfer speed can also be set by key-in operation. If you wish to set the transfer speed to 2400 baud, enter the following direct command.

POKE \square \$002B, 0.(for 2400 baud)

However, if you wish to set the transfer speed to 600 baud, enter the following direct command.

POKE_ \$002B, 1 (for 600 baud)

If you wish to load the program recorded on a cassette tape, the loading (LOAD) transfer speed is automatically set independently of the transfer speed selection switch. Be sure to store a program on and read it out from the cassette tape, using the same cassette tape recorder.

Even if the recording and playback speeds of the tape recorder are changed, the tape transport speed is not changed. However, the load/save time at 2400-baud transfer mode is one-fourth that at the 600-baud transfer mode.

Summary: Conditions for Determining Recording and Playback Speed

- (1) When the power switch is turned on after the transfer speed selection switch is set to a desired position.
- (2) When the following sentences are entered:
POKE \$002B, 0 (for 2400 baud)
POKE \$002B, 1 (for 600 baud)
- (3) When a program is loaded (LOAD) from the cassette tape.

Notes:

- (1) The recording and playback speeds may not be switch selected until the power switch is turned off.
- (2) The automatic transfer baud rate change in the playback mode does not allow resetting of the transfer speed. The preset speed is effective in the SAVE mode.
- (3) If you wish to reset the transfer speed using the transfer speed selection switch, set the transfer speed selection switch to the desired position, turn off the power switch and then turn it on. If a program is already loaded, use the POKE sentence as described in Summary (2).

3. Program Recording (SAVE)

"SAVE" is an operation for recording a program created by you on a cassette tape. Confirm the predetermined speed to be set in accordance with "2. Baud Rate Setting".

Use a cassette tape which has a short recording time, such as Technics RT-30EN (C-30) for avoiding recording trouble. Set the magnetic recording tape and reset the counter to "0".

When "Ready" is displayed, press keys to enter data as follows, and press the **RETURN** key to complete entry. (If any, delete unnecessary characters left on the row where the cursor is located.)

SAVE _ "ABC"

where "ABC" is the program name. **The program name is arbitrarily determined to have a maximum length of 16 alphanumeric characters.** (Do not use any symbols nor spaces in the program name.)

Set the tape recorder in the record mode (simultaneously press the red record and play buttons). When a remote control jack is connected to the black plug, the tape does not travel.

Press the **RETURN** key.

On the screen, the following instructions are displayed:

Push record

Writing ABC

and the tape recorder is then started.

In this manner, the save operation is initiated. When the **SAVE** operation is completed, the following message is displayed:

Ready

The tape recorder with the remote control function is stopped. (Otherwise, press the stop button.) Rewind the tape to the beginning.

The **SAVE** operation is thus completed. Check using the **VERIFY** command (refer to page 21) if the program is properly recorded.

Mark counter reading and the program name for later application when another program is to be recorded after this one on the same tape. Perform the **BREAK** operation to interrupt the **SAVE** operation.

Notes:

- (1) Set the tape recorder in the record mode before pressing the **RETURN** key when the **SAVE** operation is performed in order to avoid loss of the beginning of the program, because writing is initiated upon depression of the **RETURN** key.
- (2) Writing is initiated with a click sound upon depression of the **RETURN** key.
- (3) Be sure to use a program name with a program, for further application, in the **SAVE** mode.

4. Program Loading (LOAD)

LOAD is for placing the program in the tape recorder into internal storage in the personal computer.

Match the counter number and the corresponding program to be loaded into the computer (the previous program can overlap the desired program).

The reading rate is automatically set. The magnetic tape recorded at the rate of 2400 baud may not be read correctly if the special purpose tape recorder is not used.

When "Ready" is displayed, press keys to enter data as follows, and press the **RETURN** key to complete entry. (If any, delete unnecessary characters left on the row where the cursor is located.)

LOAD — "ABC"

where "ABC" is the program name.

Press the **RETURN** key. The following instruction is displayed.

Push play

Reproduce (PLAY) the data recorded on the cassette tape at the highest volume level. The next statement is then displayed within a short period of time:

Loading ABC

LOAD is initiated.

When the following instruction is displayed:

Ready

A tape recorder with remote control function is stopped.

Thus, program loading is completed. If the display is:

Tape read error

Ready

an error has occurred in the loading operation of the program. Turn down the volume level slightly and repeat the LOAD operation until an optimal volume level is found. (A tape recorder having a volume control whose level is indicated by numerals is recommended.)

If the cassette tape is played from the beginning and another program "DEF" is recorded before the desired program on the same cassette tape, the following statement is displayed:

Skip DEF

Only the desired program is searched.

Alternatively, when the LOAD operation only is specified without entering the program name as follows:

LOAD

The first program encountered is loaded. To cancel the LOAD operation, perform **BREAK** key operation.

Note:

- (1) First press the **CTRL** key and then **LOAD** to set the tape recorder in the **LOAD** operation. This procedure is opposite to the **SAVE** operation (in the case of a tape recorder with a remote control jack, even if the play button is pressed, the tape recorder will not start. Depress the **RETURN** key to automatically start the tape recorder).
- (2) The reading operation is initiated with a click sound upon depression of the **RETURN** key.
- (3) Set the tape recorder volume control to the optimal level. In general, the treble control is set at high level, and the volume control is set at medium level (avoid setting the volume control at maximum level).
- (4) Be sure to keep the tape recorder away from the TV monitor. Depending on the type of the tape recorder used (if it is not the recommended model) it may be necessary to only plug the Mic or the Monitor plug in at one time in order to get satisfactory saves or loads. In some tape recorders there is interaction between these jacks that can cause problems.
- (5) Some program tapes are available with 2400 baud on one side and 600 baud on the other side. If you do not have a special purpose tape recorder or a tape recorder designed for the JR-200U, only use the program tape for 600 baud. When using the special purpose tape recorder, use either side of the program tape.

5. Recording Check (VERIFY)

The **VERIFY** command is used to check if the program is properly recorded (**SAVED**) on tape. In the same manner as in the **LOAD** operation, enter:

VERIFY _ "ABC"

The program name need not be specified.

When the **VERIFY** operation is completed, the following statement is displayed:

Ready

However, if the display is:

Verify error

repeat the **SAVE** operation. During the verify operation, the program to be recorded again is retained in the memory. If you load another program without recording the program to be recorded, the first program may be lost.

The **VERIFY** operation is also possible in the **MSAVE** operation. (see next page)

Note:

When the message "Verify error" is displayed on the screen, check whether the baud rate is matched with that of the tape recorder or not.

6. Machine Language Recording and Loading (MSAVE and MLOAD)

The "MSAVE" or "MLOAD" mode allows writing and reading of machine language and data with respect to the memory. The SAVE and LOAD operations for machine language and data are the same as those for the BASIC language. However, when MSAVE operation is performed, addresses corresponding to the storage data addresses must be specified. For example, when data between addresses \$3000 and \$3100 is to be stored by the MSAVE operation. Enter:

MSAVE _ "ABC", \$3000, \$3100

where "ABC" is the file name. **The program name is arbitrarily determined to have a maximum length of 16 alphanumeric characters** (Do not use any symbols nor spaces in the program name). In order to store data by the MLOAD operation, enter:

MLOAD _ "ABC"

However, the file name "ABC" can be omitted. Since data is written in a data area corresponding to addresses \$3000 to \$3100, be careful not to overlap the BASIC program area and the data area.

Note: The dollar sign before a number denotes hexadecimal (base 16).

The start address can be changed. When you wish to write data corresponding to addresses starting from \$3500, enter:

MLOAD _ "ABC", \$3500

Data is written in the data area corresponding to addresses beginning from address \$3500 since the updated start address has the priority over the previous start address.

The MSAVE and MLOAD sentences can be used within a program. When data is once read out from a cassette tape, it remains in memory until power is OFF and any other data in these addresses is overwritten by this data.

If the above is used within a program, "Push Play" is not displayed on the screen.

User File:

Data can be written on or read out from a cassette magnetic tape as a sequential file. Refer to the BASIC sentences such as OPEN, CLOSE, PRINT# and INPUT# sentences for further details.

7. Merge of Two Programs (MERGE)

The MERGE command is used to merge two programs into a single program. Load program

A and then program B. Then, enter:

MERGE _ "B"

In this case, program B must have line numbers greater than those of program A. If the line numbers are mixed with respect to programs A and B, or if the same line numbers are present in programs A and B, the operation may not be successfully performed.

The program obtained by merging two programs is regarded as a single program.

8. Other Operations

1. Joystick

Two joysticks can be connected to the main unit.

Use the following data input sentence for joystick operation.

A = STICK (n) where n is 0, 1 or 2

When n = 0 is given, a character code corresponding to the key which is currently pressed is given to variable A (in the same manner as the PICK sentence).

When n = 1 is given, the joystick (no. 1) is operative.

When n = 2 is given, the joystick (no. 2) is operative.

"A" above takes the following values corresponding to the conditions of the joystick.

| Conditions | Computation of Variable A | Input |
|---------------------------------------|---------------------------|----------|
| Neutral | 255 | 11111111 |
| Upper | $255 - 1$ | 11111110 |
| Lower | $255 - 2$ | 11111101 |
| Left | $255 - 4$ | 11111011 |
| Right | $255 - 8$ | 11110111 |
| Upper left | $255 - 4 - 1$ | 11111010 |
| Lower left | $255 - 4 - 2$ | 11111001 |
| Upper right | $255 - 8 - 1$ | 11110110 |
| Lower right | $255 - 8 - 2$ | 11110101 |
| Switch depression at neutral position | $255 - 16$ | 11101111 |

For example, when you press the switch while pulling the joystick to the lower right, variable A takes $255 - 16 - 8 - 1 = 229$.

2. Special Operations

The following operations can be performed using the following commands.

| Sentence | Execution Result |
|--|--|
| POKE $_ \$CA00, n$ n: the color code | The frame color is changed in accordance with the specified color code (the frame color is black immediately after the power switch is turned on). |
| POKE $_ \$002B, 0$ | The cassette tape recording (SAVE) speed is switched to 2400BPS (baud). |
| POKE $_ \$002B, 1$ | The cassette tape recording (SAVE) speed is switch to 600BPS (baud). |
| POKE $_ 0, 0$ POKE $_ 0, \$40$ | No buzzer sound is produced even if a key is pressed. Buzzer sounds are produced when a key is pressed. |
| A = PEEK($\$0030$) B = PEEK($\0031) | A musical note address (musical note length code) of the first music piece currently performed is obtained by the expression $256 \times A + B$ (used for screen synchronization). When this operation is completed, the value becomes 0. |
| A = PEEK($\$00D9$) B = PEEK($\$00DA$) C = PEEK($\$00DB$) D = PEEK($\$00DC$) | The address range of data after the MLOAD operation falls in a range between $(256 \times A + B)$ and $(256 \times C + D)$. |
| CTRL + SHIFT +] CTRL + SHIFT + : | The BREAK key operation will be cancelled. The BREAK key operation will be restored. |

3. Key Input

When you press a key, the key signal is temporarily stored in a buffer and is transferred to the CPU. When you prestore data, to be used later even during operation of the CPU, such data can be fetched in the CPU as needed. In this operation, the corresponding characters are not displayed on the screen immediately when you press the keys. However, when they are fetched by the CPU, the corresponding characters are displayed on the screen. For example, when you subsequently enter the LOAD and MLOAD statements, or when you enter data using the INPUT sentence, the overall operating time can be saved.

However, the PICK statement is an exception. When you enter a PICK statement, data stored in the buffer is cancelled, while the data of the key currently pressed becomes effective. Furthermore, when the BREAK key operation is performed, and when an error occurs, the storage contents in the buffer are cancelled.

8. SPECIFICATIONS OF THE BASIC LANGUAGE

The general description of JR-BASIC 5.0 (to be referred to as a simply BASIC hereafter) will be made. The syntax of BASIC can be classified into commands, statements and functions.

The commands, statements (sentences) and functions are called the machine language. The commands are used to edit the program in the command modes, but cannot be used in the BASIC sentences. The commands are:

AUTO, CONT, DELETE, FIND, LFIND, LIST, LLIST, MERGE, MON, and NEW.

The statements are used in the program to describe the steps of a flow chart by the machine language. Unless otherwise specified, the statements can be used in the command mode.

The functions are used to represent numerical values and data.

1. Program Format

Every BASIC program consists of lines, and each line is regarded as a basic unit in the program:

Line number <BASIC sentence> [: <BASIC sentence> . . .] **RETURN**

BASIC sentences (statements) can be written on one line by being separated by a colon. Each line starts with a line number and is ended by pressing the **RETURN** key.

An integer between 1 and 32767 is selected for the line number. The functional character should be divided by a space or the symbol (":", ",", ";" etc.). The functional characters or variable names which are input with the small characters are converted into the large characters. Eighty characters are input in each row except for **RETURN**. The program is easy to read if the spaces are used. However, the spaces are not available in the following cases.

- In the line number
- In the spelling of a functional character (key word)
- In the spelling of a variable name
- In a value constant
- In a comparison operator between two or more characters.

2. Variables

The variables are classified in accordance with the types of values or character-strings.

Variables:

- Value variable Letter (number)
Simple variable A, AB, X2
Array variable A(1), AB(5), A1(4, 3)
- Character-string variable Letter (number) \$
Simple character-string variable A\$, AB\$, A2\$
Array character-string variable A\$(3), X2\$(I, J)

Variables with the same name as a functional character (key word) is not permitted.

The array variable or array character-string variable is selected in a single or two-dimensional array, and the lowest limit of the subscript value is 0. The single character-string variable consists of data of 32 characters at maximum.

During execution of the program, one value variable corresponds to one value, and one character-string variable corresponds to one character array. Any variable can be substituted by any value or character-string, using a substitution sentence. An array variable refers to an element selected by a subscript value among one- and two-dimensional arrays. If the same letter is used for the value and character-string variables, no relationship is present between the value and character-string variables.

3. Constants

Constants consist of value constants and character-string constants.

Constants:

- Value constants

| | | |
|-----------------------------|--------------------------|---------------|
| Integer type | sd . . . d | 1. -21 |
| Fixed decimal point type | sd . . . drd . . . d | 3.14, .255 |
| Floating decimal point type | sd . . . drd . . . dEsdd | 6E23, 1.6E-19 |
| Hexadecimal type | \$dddd | \$41, \$C100 |
- Character-string constants " " or ' ' "JR-200 _ BASIC", 'ABC'

where s: + or - (+ can be omitted), d: number; r: decimal point; and E: power of 10 (example: E-19 means 10^{-19})

Constants are used with a BASIC program without any modification. A constant has a maximum length of nine digits (decimal notation). The range of the expression is between 10^{-39} and 10^{+38} . A constant smaller than the minimum value is substituted by 0.

Character-string constants are within quotation marks " " or ' ', and consist of 73 characters at maximum. A space is counted as a character.

4. Expressions and Operations

Expressions consist of value expressions and character-string expressions. The value expressions include value constants, value variables, value functions and value elements which are connected by operators. The operated results are all values. Character-string expressions are character-string constants, character-string values, character-string functions, and character-string elements which are connected by operators. The operated results are all character-strings.

(1) Arithmetic Expressions and Operations

The construction of value expressions and operations basically comply with the rules of algebra. Operations include the following types:

arithmetic operations, comparison operations, logic operations, and function operations.

1. Arithmetic Operations

The arithmetic operators are shown below:

| | | |
|-----------|--------------------------|-------------------|
| \wedge | power (exponent) | $X \wedge Y$ |
| $-$ | negative | $-X$ |
| $*$, $/$ | multiplication, division | $X * Y$, X / Y |
| $+$, $-$ | addition, subtraction | $X + Y$, $X - Y$ |

Calculation priority is defined as follows: power, negative, multiplication and division, and addition and subtraction. However, when parentheses are present, the elements within the parentheses are first calculated.

In the case such as $X \wedge Y$, the following results are specified.

| X | Y | results |
|----------|-------------|---------|
| 0 | 0 | 1 |
| positive | 0 | 1 |
| 0 | negative | error |
| negative | real number | error |

2. Comparison Operations

A comparison operation compares two different value data or character-string data and returns a value which is true (1) or false (0). The comparison operators are classified as follows:

| | | |
|-----|--------------------------|-----------|
| = | equal to | $X = Y$ |
| < > | not equal to | $X < > Y$ |
| < | less than | $X < Y$ |
| > | greater than | $X > Y$ |
| < = | less than or equal to | $X < = Y$ |
| > = | greater than or equal to | $X > = Y$ |

3. Logic Operations

A logic operation has a value which is true (1) or false (0).

| Operational precedence | Logical operator | meaning |
|------------------------|------------------|----------|
| 1 | NOT | negative |
| 2 | AND | and |
| 3 | OR | or |

Logic expression for "exclusive or", that is, XOR is obtained as follows:

IF $\underline{\quad}$ $(A < B) + (C > D) = 1$ $\underline{\quad}$ THEN $\underline{\quad}$ the line number of the statement.

4. Function Operations

Functions are used to call a predetermined arithmetic operation in the expression.

27 functions are available.

5. Operational Priority

The operational priority is defined as follows:

1. Arithmetic operations
2. Comparison operations
3. Logic operations

(2) Character-string Expressions and Operations

Two types of character-string expressions are available: a character-string expression obtained by combining characters by the addition operator; and a character-string expression obtained by a comparison operator.

1. Character-String Expression Combined by the Operator "+"

"ABCD" + "EFGH" → "ABCDEFGH"

If the calculated result obtained by combining as described above is more than the maximum value, the overflowed characters are neglected.

2. Comparison of Character-String Expressions by a Comparison Operator

The comparison of the character-string data complies with the following rules.

- Each of the characters of a character-string expression is compared with each of the characters of another character-string expression from the beginning. (A space between the characters is regarded as a blank character.)
- Characters are compared with each other in accordance with the character codes.

5. Array Declaration and Element Reference

An array which is a set of data can be accessed by an array variable name. The array variable name declares the name of the array and the number of elements included in this set of data. The array declaration has a format as follows:

`DIM_ <Array Variable Name> (<Limit 1> [, <Limit 2>]) [, <Array Variable Name>
<Limit 3> [, <Limit 4>]), ...]`

When limit 2 is omitted, the format is called a one-dimensional array; when limits 1 and 2 are specified, the format is called a two-dimensional array. The values of limits 1 and 2 indicate the maximum number of elements — 1, which may be selected within the allowable range of the memory from 0.

(Example)

When array X having 10 elements is declared:

`DIM X(9)`

When array Y having $5 \times 4 = 20$ elements is declared:

`DIM Y (4, 3)`

An element of the array is specified by an array variable. The format of the array element is as follows:

`<Array Variable Name> (<Expression 1> [, <Expression 2>])`

where the lower limit of expressions 1 and 2 is 1 and the upper limits of the expressions 1 and 2 are the values of limits 1 and 2, respectively.

For Format Reference:


- Proper formats for commands and statements are described in the format of the BASIC language in accordance with the following descriptive rules.
 - (1) Input an item represented by letters as it is.
 - (2) Specify the contents of items with the symbols $\langle \rangle$.

When the contents within the symbols represent a description such as "relation" or "relation n", the relation here means an arithmetic relation.
 - (3) Omit items with the symbols $[]$ since they are optional items.
 - (4) Be sure to select one element of the contents of an item with the symbols $\{ \}$, since these are selective.
 - (5) The symbol $|$ indicates "or".
 - (6) The symbol \dots indicates repetition, within the range of one line of the item immediately before the current item.
 - (7) Input symbols such as $()$, $"$, $'$, $, , ;$, $:$ and $_$ (space) as they are.
 - (8) Do not input symbols such as $[]$, $\langle \rangle$, $\{ \}$, $|$ and \dots .
- The key words indicated by a shaded rectangle can be entered by a single key.

6. BASIC Commands and Statements (Sentences)

| No. | Name | Format and Function | Example and Summary |
|-----|------------------------------|---|--|
| 1 | AUTO | <p>AUTO <code>[_<Line number>[!, -<Increment>]]</code></p> <ul style="list-style-type: none"> To automatically generate a line number and to sequentially display the line numbers. The line number is from 1 to 32767. | <p>AUTO: The line number begins from 10 and every tenth line number is displayed.</p> <p>AUTO_100: The line number begins with 100 and every tenth line number is displayed.</p> <p>AUTO_100, 100: The line number begins from 100 and every hundredth line number is displayed.</p> <p>AUTO_, 100: The line number begins from 10 and every hundredth line is displayed.</p> |
| 2 | BEEP | <p>BEEP <code>[_!0 1]</code></p> <ul style="list-style-type: none"> To control ON/OFF of the tone. When only BEEP is specified, the tone is produced for about 0.5 second. | <p>BEEP_1: Tone is generated.</p> <p>BEEP_0: Tone is not produced.</p> |
| 3 | CLEAR | <p>CLEAR <code>[_<Expression>]</code></p> <p>To delete all the variables and to limit the end address of the memory used in the BASIC formats.</p> | <p>CLEAR: All the variables are deleted.</p> <p>CLEAR_\$7000: All the variables are deleted and to set the upper limit address to \$7000.</p> |
| 4 | CLOSE | <p>CLOSE <code>[_!0 1]</code></p> <p>To declare the end of the file which has used.</p> | <p>The EOF (end of file) bit is set by this command.</p> |
| 5 | CLS (Clear screen) | <p>CLS</p> <p>To delete currently displayed contents on the screen and to move the cursor to the home position (upper left corner of the screen).</p> | <p>When CLS is used within the program, one key command can not be used for input.</p> <p>You must input C, L, S.</p> |
| 6 | COLOR | <p>COLOR <code>_[(Expression 1)] [, [(Expression 2)] [, (Expression 3)]]</code></p> <p>To specify the colors of characters and the background, color inversion of the characters and the background, and a user's definition character.</p> | <p>Expression 1: The color of characters and graphics are specified. (Refer to the color code table.)</p> <p>Expression 2: The color of the background of the screen is specified. (Refer to the color code table.)</p> <p>Expression 3: The display mode is selected as follows.</p> <p>0: normal mode; 1: user's definition mode; 2: inversion mode; 3: background color change mode.</p> |

| No. | Name | Format and Function | Example and Summary |
|-----|--------------------|--|--|
| 6 | COLOR | | <p>Mode 0: The user's definition mode is released.</p> <p>Mode 1: The user's definition character can be displayed.</p> <p>Mode 2: The character color and the background color is inverted.</p> <p>Mode 3: The background color from the current position of the cursor to the end of the screen is changed into the effective color specified by Expression 2.</p> |
| 7 | CONT (Continue) | <p>CONT</p> <p>To restart a program which is interrupted by the STOP sentence or BREAK mode (CTRL + C key).</p> | <p>Execution:</p> <p>RUN↓</p> <p>Interruption: STOP (or CTRL + C key) direct check the program.</p> <p>↓</p> <p>Restart:</p> <p>CONT↓</p> |
| 8 | DATA | <p>DATA_⟨Constant⟩ [, ⟨Constant⟩ ...]</p> <p>To store data in the data pool.</p> | <p>50_DATA_10, 5</p> <p>100_DATA_ABCD, 0</p> <p>The above example is the same as the DATA sentence DATA 10, 5, ABCD, 0.</p> <p>200_DATA_"_ABC_",</p> <p>"DE, FG", HIJKL</p> |
| 9 | DELETE | <p>DELETE_⟨[Line no. 1]⟩ [!,]⟨[Line no. 2]⟩</p> <p>To delete a specific line.</p> | <p>DELETE_100: Line 100 is deleted.</p> <p>DELETE_100,: The lines after line 100 are all deleted.</p> <p>DELETE_100-: The lines after line 100 are all deleted.</p> <p>DELETE_, 100: The lines from the beginning of the program to line 100 are deleted.</p> <p>DELETE_100-300: The lines between line 100 and line 300 are deleted.</p> |
| 10 | DIM (Dimension) | <p>DIM_⟨Array variable name⟩ (⟨Expression 1⟩ [, ⟨Expression 2⟩]) [, ⟨Array variable name⟩ (⟨Expression 3⟩ [, ⟨Expression 4⟩]), ...]</p> <ul style="list-style-type: none"> • The length of the array variable name is specified to secure its area. • The values of ⟨Expression 1⟩ and ⟨Expression 2⟩ to be declared start from 0. | <p>DIM_A (10): The number of elements of one-dimensional array variable name A is 11. (0 to 10)</p> <p>DIM_B (5,3): The number of elements of two-dimensional array variable name B is 24.</p> <p>DIM_C\$(10)*15: The number of elements of one-dimensional character-string variable name is 11,</p> |

| No. | Name | Format and Function | Example and Summary |
|-----|---------------------------|--|--|
| 10 | DIM (Dimension) | <ul style="list-style-type: none"> The value of the expression is defined by the declaration of the character-string variable, within the range from 1 to 255 at maximum. | and the number of characters of each element is 15 at maximum. DIM_D\$*80: The number of character of character-string variable D\$ is 80 at maximum. |
| 11 | END | END To complete execution of a program and to return to the command level. | |
| 12 | FIND | FIND _ "Character-string" [, [(Line no. 1)] [, - [(Line no. 2)]] To search for a line including a specified character-string and to display the entire line. This command may be conveniently used for selective display of a line in debugging. | Search for the line including character-string PRINT FIND _ "PRINT": The entire program is searched. FIND _ "PRINT",100: Only line 100 is searched. FIND _ "PRINT",,100: All the lines between the beginning of the program and the line number 100 are searched. FIND _ "PRINT", 100,: All the lines after line 100 are searched. FIND _ "PRINT",100, 500: The lines from line 100 to line 500 are searched. |
| 13 | FOR – NEXT | FOR _ (Value variable) = (Expression 1) _ TO _ (Expression 2) [_ STEP _ (Expression 3)] STEP 1 can be omitted. NEXT _ [(Value variable)] <ul style="list-style-type: none"> To perform a sentence between FOR and NEXT sentences as many times as specified. A multiloop can be also performed. | Multiplication table operation 10_ FOR _ X = 1_ TO _ 9 20_ FOR _ Y = 1_ TO _ 9 30_ PRINT _ X; "X"; Y; "="; X*Y, 40_ NEXT _ Y 50_ PRINT 60_ NEXT _ X  |
| 14 | GOSUB – RETURN | GOSUB _ (Line number) RETURN To access a subroutine and to return to the next line. | A subroutine beginning from line number 500 is accessed. 100_ GOSUB _ 500 500_ REM _ SUBROUTINE 600_ RETURN |
| 15 | GOTO | GOTO _ (Line number) To jump to a sentence whose line number is specified. | 100_ GOTO _ 200 To jump to line number 200 200_ PRINT _ "JR-200" |

| No. | Name | Format and Function | Example and Summary |
|-----|--------------------------------|---|---|
| 16 | HCOPY (Hard copy) | HCOPY <ul style="list-style-type: none"> To print the currently displayed image on the screen. Use the BREAK key operation to interrupt printing. | |
| 17 | IF — THEN | IF _ (Expression 1) Comparison operator or Logic operator (Expression 2) _ THEN _ line number To control the program sequence in accordance with the conditions specified by a comparison expression. | IF _A <= 0 _ THEN _ GOTO _ 100 (GOTO immediately after THEN may be omitted.) If A is zero or negative, GOTO 100 is executed. |
| 18 | INITP | INITP To perform the initial setting of the printer. | |
| 19 | INPUT | INPUT _ [“(Prompt sentence)” ; ,] (Variable) [, (Variable) ...] Prompt sentence = “(Character-string)” <ul style="list-style-type: none"> To input data with the keyboard and to substitute the data for a variable. When this sentence is input, the ? mark is displayed on the screen to request the data to be input. | 10 _ INPUT _A ,B ,C\$ 20 _ PRINT _A ,B ,C\$ RUN ? 5, -10, JR-200 RETURN key is pressed. 5 -10 JR-200 are displayed on the screen. |
| 20 | INPUT# (Input sharp) | INPUT# _ { 0 1 } , [(EOF _ line no.)] (Variable) [, (Variable) ...] <ul style="list-style-type: none"> To output the data from a file and to substitute the data for a variable. To jump to the specified line number upon the end of the file (EOF). | When a long character data is substituted for a character-string variable, the overflowed characters are neglected. When the file number is other than 0 and 1, an error (Value error) occurs. When the file number is not declared by the Open sentence, an error (File access error) occurs. |
| 21 | LET | [LET _] (Variable name) = { (Arithmetic expression) (Character-string expression) } The key word LET may be omitted, it is optional. <ul style="list-style-type: none"> To evaluate an expression and to replace a variable with the calculated results. | 10 _ LET _I = 5 (5 is substituted for I.) 20 _ LET _J = I + 3 (8 is substituted for J.) 30 _ LET _K = I * I + J (33 is substituted for K.) 40 _ LET _A\$ = “JR-200 _” 50 _ PRINT _I , J 60 _ PRINT _K , A\$ RUN 5 8 33 JR-200 |

| No. | Name | Format and Function | Example and Summary |
|-----|--------|--|---|
| 22 | LFIND | <p>LFIND_ "Character-string" [, [(Line No.1) [{, - } [(Line No.2)]]]</p> <p>where [LFIND_ "Character-string" ,] is not available.</p> <p>To output the line including the character-string to a sheet of paper in the printer.</p> | |
| 23 | LIST | <p>LIST [_(Line No.1) [{, - } (Line No.2)]]</p> <p>To display all or part of a program stored in the memory.</p> | <p>LIST: The entire program is displayed.</p> <p>LIST_100: Only line number 100 is displayed.</p> <p>LIST_100, 200: Contents between line numbers 100 and 200 are displayed.</p> <p>LIST_100,: Contents after the line number 100 are displayed.</p> <p>LIST_,100: Contents between the beginning of the program and the line number 100 are displayed.</p> |
| 24 | LLIST | <p>LLIST[_[(Line No.1) [{, - } Line No.2)]]</p> <p>To print all or part of the program, stored in the memory, to the printer.</p> | <p>LLIST_100, 200: Contents between line numbers 100 and 200 are printed.</p> |
| 25 | LOAD | <p>LOAD[_"File name"]</p> <p>where File name is a character-string of up to 16 characters.</p> <ul style="list-style-type: none"> ● To retrieve a BASIC program from the cassette tape and to store it in the memory. ● The baud rate is automatically set. | <p>LOAD_"GAME" ← (which indicates that file name GAME is input.)</p> <p>LOAD ← (which indicates that the first file found from the current position of the cassette tape is input.)</p> |
| 26 | LOCATE | <p>LOCATE_ (Expression 1), (Expression 2)</p> <p>Expression 1 . . . abscissa ($0 \leq \text{Expression 1} \leq 31$).</p> <p>Expression 2 . . . ordinate ($0 \leq \text{Expression 2} \leq 23$).</p> <ul style="list-style-type: none"> ● To move the cursor to an arbitrary position on the screen. ● To move the cursor to an arbitrary position within the following range; the upper left corner is defined as the origin (0, 0) and the lower right corner is defined as (23, 31). | <p>The heart mark "♥" is to be displayed on Line 5, Digit 13:</p> <p>LOCATE_13, 5:PRINT "♥"</p> <p>A line is to be drawn from Line 2 to Line 10, Digit 8.</p> <pre> 10_FOR_I = 2_TO_10 20_LOCATE_8, I 30_PRINT_" " 40_NEXT_I </pre> |
| 27 | LPRINT | <p>LPRINT_(Print element) [(Division symbol) [(Print element) . . .]</p> <p>To print value data or character-string data at the printer.</p> | <p>This command is the same as the PRINT command except that the results are printed out at the printer.</p> |

| No. | Name | Format and Function | Example and Summary |
|-----|--------------------------------|---|--|
| 28 | MERGE | <p>MERGE[_"File name"]</p> <ul style="list-style-type: none"> ● To add to the BASIC file specified in the program memory (combining of two programs). ● To merge a program having smaller line numbers into a program having larger line numbers, the latter program is read out from the cassette tape. | <p>When newly added line numbers of the program are smaller than those of the previous program, the operation may not successfully be performed.</p> <p>This command is the same as the LOAD command except for the condition described above.</p> |
| 29 | MLOAD (Machine load) | <p>MLOAD[_"File name"] [, <Expression>]</p> <p>To retrieve a machine language (program) recorded on a cassette tape and to store it in memory.</p> | <p><Expression> indicates an initial loading address.</p> <p>When <Expression> is omitted, data from the start address specified by the MSAVE command is loaded.</p> <p>When <File name> is omitted, the file found first is loaded (a file saved by the SAVE command is skipped).</p> |
| 30 | MON (Monitor) | <p>MON</p> <p>To give control to the built-in machine language monitor.</p> | <p>The command removes the control from the command level of BASIC to the machine monitor. It is convenient to input or correct a machine language program.</p> |
| 31 | MSAVE (Machine Save) | <p>MSAVE _"File name" , <Initial address> , <End address></p> <p>where File name is a character-string of up to 16 characters.</p> <p>To put a file name for a machine language program stored in the memory and to record it on a cassette tape.</p> | <p>Address is decimal, or hexadecimal with \$ mark. The file name is a character-string (alphanumeric) up to 16 characters.</p> |
| 32 | NEW | <p>NEW</p> <p>To delete all the BASIC programs stored in the memory.</p> | |
| 33 | ON – GOSUB | <p>ON _ <Expression> _ GOSUB _ <Line No. 1> [, <Line No. 2> . . .]</p> <p>To jump to the line specified by the expression value.</p> | <p>50 _ ON _ J _ GOSUB _ 100, 200, 300</p> <p>When J = 3, jump to line number 300</p> |
| 34 | ON – GOTO | <p>ON _ <Expression> _ GOTO _ <Line No. 1> [, <Line No. 2> . . .]</p> <p>To jump to the line specified by the expression value.</p> | <p>50 _ ON _ J _ GOTO _ 100, 200, 300</p> <p>When J = 3, jump to line number 300</p> |

| No. | Name | Format and Function | Example and Summary |
|-----|-------------------------|---|---|
| 35 | OPEN | <p>OPEN_ " { I O } ", { 0 1 }, "<File name>"</p> <ul style="list-style-type: none"> • "I" mode ... input from the file (already registered). • "O" mode ... output to the file (newly registered). • File Name is a character-string within the range of 16 characters. <p>To declare the start of the file so as to allow file editing.</p> | <p>Be sure to open the file by an OPEN sentence before an INPUT# sentence or a PRINT# sentence are used.</p> |
| 36 | PICK | <p>PICK_<Value variable></p> <p>To replace, in decimal notation, a variable by the character code of a key which is pressed when this PICK command is performed.</p> | <p>For waiting while looping until a key is pressed.</p> <pre>10_PICK_K 20_IF_K = 0_THEN_10</pre> |
| 37 | PLAY | <p>PLAY_ " { B F } ", [<Expression 1>] [, <Expression 2>] [, <Expression 3>]</p> <p>"B" ... Background "F" ... Foreground</p> <ul style="list-style-type: none"> • A triad chord can be performed by three expressions: a musical piece is performed simultaneously while BASIC operation is performed. | <p><Expression 1>, <Expression 2>, and <Expression 3> respectively indicate start addresses of musical note data.</p> <p>When a SOUND or BEEP sentence is entered while a musical piece is being performed as a background music, musical performance specified by <Expression 1> is stopped.</p> |
| 38 | PLOT | <p>PLOT_<Expression 1>, <Expression 2></p> <ul style="list-style-type: none"> • To display 4 dots x 4 dots at the specified position on the screen. • Use a COLOR sentence to specify a desired color. | <p>Display Four red points as follows:</p> <pre>10_CLS:COLOR_2 20_PLOT_10, 10 30_PLOT_53, 10 40_PLOT_10, 37 50_PLOT_53, 37</pre> |
| 39 | POKE | <p>POKE_<Expression 1>, <Expression 2></p> <ul style="list-style-type: none"> • To write data of 1 byte in a specified address of the memory. • To write the lower byte of <Expression 2> to the memory address of <Expression 1>. | <p>For writing data of 5 in address \$1000 of the memory.</p> <pre>POKE_\$1000, 5</pre> <p>Be careful not to destroy data of the memory area which is used for the BASIC language in accordance with the memory map.</p> |
| 40 | PRINT | <p>PRINT <Display element> [<Division sign> <Display element> ...]</p> <p>To display value and character-string data on the display screen.</p> | <pre>10_A = 123 20_PRINT "A="; A; "M" RUN A = _123_M ↑ ↑ (space)</pre> |
| 41 | PRINT# (Print sharp) | <p>PRINT#_ { 0 1 } [<Expression arrangement>]</p> <p>Output data to a file.</p> | <p>The file data is output to the file number specified by OPEN sentences.</p> |

| No. | Name | Format and Function | Example and Summary |
|-----|------------------------|---|---|
| 42 | RANDOMIZE | RANDOMIZE To change the value of RND function into a random value. | 10_A = INT (RND(1) * 6) 20_RANDOMIZE 30_B = INT (RND(1) * 6) |
| 43 | READ | READ_ <Variable> [, <Variable> ...] <ul style="list-style-type: none"> ● To read out data from the data pool and substitute it for a variable. ● Use the RESTORE sentence when the readout data is to be read out again. | 10_DATA_1, 2, 3 20_DATA_"DATA" 30_READ_A, B 40_READ_C, D\$ |
| 44 | REM (Remark) | REM [_ <Character-string>] <ul style="list-style-type: none"> ● To describe remarks in a program. ● The REM sentence has no function in performance of the program. | |
| 45 | RESTORE | RESTORE [_ <Line number>] <ul style="list-style-type: none"> ● To map the data readout position of the data pool to the beginning of the data pool or with the beginning of an arbitrary DATA sentence. ● The RESTORE sentence may be repeated. | 10_READ_A, B 20_RESTORE 30_READ_I, J 40_RESTORE_70 50_READ_X, Y 60_DATA_1, 2, 3, 4 70_DATA_-1, -2, -3, -4 |
| 46 | RUN | RUN [_ "<File name>" <Line number>] <ul style="list-style-type: none"> ● To begin the program. ● The values of all variables are entirely reset. ● The cursor is not displayed in this mode. | RUN_"NEXTFILE" The program is loaded from the cassette tape recorder and run. RUN The program is run from the beginning. RUN 1000 The program is run from line number 1000. |
| 47 | SAVE | SAVE_"<File name>" where the file name is the character-string within the range of 16 characters. <ul style="list-style-type: none"> ● To record a BASIC program with a file name stored in the memory. | SAVE_"GAME1" ← File name GAME1 is assigned to the program and this program is recorded on the cassette tape. |
| 48 | SOUND | SOUND_ <Expression 1>, <Expression 2> where <Expression 1> is setting value of a frequency. 0 to 65535 (Hz) and <Expression 2> is described in "TEMPO". To produce sounds at the speaker at a specified frequency for a specified time interval. | SOUND_1000, 50 A sound of 1 kHz is continuously produced for 1 sec (tempo = 109). |

| No. | Name | Format and Function | Example and Summary |
|-----|---------------|---|--|
| 49 | STOP | STOP To interrupt execution of a program and to restore the command mode. | <Program> <Command> 10_ A = 5 ← RUN to 50_ STOP ← Break in line 50 PRINT_ A (To display A). A = 3 (To substitute 3 for A). CONT ← Restart. |
| 50 | TEMPO | TEMPO_ <Expression> where the initial value of <Expression> is 109, which corresponds to the tempo 120(= 120), and value 50 is corresponds to approximately 1 second using SOUND sentence. If you wish to change the tempo, use the following equation: Value of <Expression> = (120 x 109)/A. To specify the reference tempo and the note length (musical note and rest) using PLAY sentences respectively. | If you wish to determine the number of quarter notes per one minute to be 60, enter: TEMPO_218 (120/60 x 109 = 218) |
| 51 | VERIFY | VERIFY [_" <File name>"] To verify whether or not a program is recorded correctly on the cassette tape in the SAVE or MSAVE command. | VERIFY_ "GAME1" The file with file name GAME1 is read out and compared with the corresponding contents in the memory. |

7. BASIC Functions

| No. | Name | Format and Function | Example and Summary |
|-----|----------------------------------|---|--|
| 1 | ABS (Absolute) | ABS (<Expression>) To give the absolute value of the expression. | (A - B) / C : ABS ((A - B)/C) |
| 2 | ASC (ASCII) | ASC (<Character-string Expression>) To give, in decimal notation, the character code of the first character of the character-string data. | 10_ A\$ = "0123" 20_ PRINT_ ASC(A\$) 48 (decimal number of character code \$38 for 0). 30_ IF_ ASC(A\$)=\$30_ THEN_ 100 To jump the line number 100. |
| 3 | CHR\$ (Character Code) | CHR\$ (<Expression>) To give a character corresponding to the character code, representing the value of the expression. | 10_ FOR_ I=48_ TO_ 57 20_ PRINT_ CHR\$(I); 30_ NEXT_ I (Display as 0123456789) |

| No. | Name | Format and Function | Example and Summary |
|-----|---|--|---|
| 4 | COS (Cosine) | COS (<Expression>) To give a Cosine value in a trigonometric function. | 10_P=3.14159265 20_PRINT_COS(P/3) (To calculate COS 60°) The unit of <Expression> value is in radians. |
| 5 | EXP (Exponent) | EXP (<Expression>) To give the value in which natural number e (e: 2.718281 ...) is powered by Expression value. | A = EXP (B) (To give e ^B to A) |
| 6 | FRE (Free) | FRE (<Expression>) To give the size of the memory area in units of bytes. | FRE(0) The memory space exception for BASIC program. FRE (anything but 0) → The memory space exception for BASIC program and Variable areas. |
| 7 | HEX\$ (Hexadecimal) | HEX\$ (<Expression>) To convert the value of the expression to 2- or 4-digit hexadecimal number and to substitute the converted value for the character-string. | 10_I=100 20_PRINT_HEX\$(I) (To display as 64) |
| 8 | HPOS, VPOS (Horizontal Position) (Vertical Position) | HPOS (<Expression>) VPOS (<Expression>) The value of the Expression has no meaning. To specify the horizontal position (HPOS) and the vertical position (VPOS) of the cursor. | 10_A=123 20_PRINT_"A";A; 30_LOCATE_HPOS(0)-1, VPOS(0) 40_PRINT_"M" (To display as A= 123M) |
| 9 | INT (Integer) | INT (<Expression>) To give a maximum integer less than the Expression value. | PRINT_INT(3.14) (To display _3) |
| 10 | LEFT\$ (Left \$string) | LEFT\$ (<Character-string variable>, <Expression>) 0 ≤ (<Expression>) ≤ 255 To extract and give the value of the character-string of the expression from the start of the character-string data stored in the character-string variable. | 10_C\$="ABC_DEF" 20_PRINT_LEFT\$(C\$, 5) (To display ABC_D) |
| 11 | LEN (Length) | LEN (<Character-string variable>) To give the number of characters stored in the character-string variable. Note: A space is a blank character. | 10_A\$="ABC_" 20_PRINT_LEN(A\$) (To display as _4) 30_PRINT_LEN(A\$ + "X5") (To display as _6) |
| 12 | LOG (Logarithm) | LOG (<Expression>) To give log of Expression value. | A=LOG (10) :PRINT_A (To give log (10) to A and to display.) |

| No. | Name | Format and Function | Example and Summary |
|-----|------------------------------------|---|--|
| 13 | MID\$ (Mid \$string) | <p>MID\$ (<Character-string variable>, <Expression 1> [, <Expression 2>])</p> <p>Expression 1 $\rightarrow 1 \leq$ <Expression 1> ≤ 255 (Position of first character)</p> <p>Expression 2 $\rightarrow 0 \leq$ <Expression 2> ≤ 255 (Number of characters to be selected)</p> <p>To give character-string data stored in the character-string variable, a desired position of which is extracted.</p> | <pre>10 A\$="PANASONIC" 20 PRINT MID\$(A\$, 3, 3) (To display NAS)</pre> |
| 14 | MOD (Modulus) | <p>MOD (<Expression 1>, <Expression 2>)</p> <p>To give the remainder of a division operation.</p> | <pre>C = MOD (A + B, 2) (For obtaining the remainder of division of (A + B) by 2.) For making a jump to a line number 100, if the value of A is an odd num- ber. 50 IF MOD(A,2)=1 THEN 100</pre> |
| 15 | PEEK | <p>PEEK (<Expression>)</p> <p>To provide data of 1 byte which is read out from the memory area corresponding to the specified memory address.</p> | <pre>For obtaining a total sum of data from address \$1000 to address \$100F of the memory. 10 S = 0 20 FOR I = \$1000 TO \$100F 30 S = S + PEEK(I) 40 NEXT I 50 PRINT S</pre> |
| 16 | RIGHT\$ (Right \$string) | <p>RIGHT\$ (<Character-string variable>, <Expression>)</p> <p>Expression $\rightarrow 0 \leq$ <Expression> ≤ 255</p> <p>To extract the number of the characters from the left end of the character-string data stored in the character-string variable.</p> | <pre>10 C\$ = "JR-200 BASIC" 20 PRINT RIGHT\$(C\$, 5) (To display BASIC)</pre> |
| 17 | RND (Random) | <p>RND (<Expression>)</p> <p>To generate random numbers within the range of 0 to 1.</p> <p>(The value for expression does not have any meaning.)</p> | <pre>For generating random numbers from 1 to 6 10 FOR I = 1 TO 6 20 PRINT INT(RND(1) * 6 + 1); 30 NEXT I (4 5 2 3 6 2: For generating random numbers 6 times)</pre> |
| 18 | SGN (Sign) | <p>SGN (<Expression>)</p> <p>To examine the sign of the value and give a number representing it.</p> <p>If (Expression) > 0 it returns 1</p> <p>If (Expression) = 0 it returns 0</p> <p>If (Expression) < 0 it returns -1</p> | <pre>ON SGN(A) + 2 GOTO 100, 200, 300 For jump to line number 100 if A is negative; to line number 200 if A is 0; and to line number 300 if A is posi- tive.</pre> |

| No. | Name | Format and Function | Example and Summary |
|-----|-----------------------------|--|--|
| 19 | SIN (Sine) | SIN (<Expression> To give a sinusoidal value of the trigonometric function. | 10_P = 3.14159265 20_PRINT_SIN (P / 6) (For calculating SIN 30°) |
| 20 | SPC (Space) | SPC (<Expression> Expression $\rightarrow 0 \leq \langle \text{Expression} \rangle \leq 255$ To display and print a specified number of spaces. | 10_FOR_I = 1_TO_5 20_PRINT_SPC(I) ; "A" ; SPC ((5 - I) * 2) ; "B" 30_NEXT_I |
| 21 | SQR (Square Root) | SQR (<Expression> To give the square root of the expression value. | 10_A = 9 : B = 16 20_C = SQR (A + B) 30_PRINT_"SQR (A + B) =" ; C (For displaying SQR (A + B) = _5) |
| 22 | STICK | STICK (<Expression> Expression \rightarrow Specify any number from among 0, 1, and 2. To get data from the joystick. | <Expression> value 0: data from the keyboard 1: data from joystick 1 (front connector position) 2: data from joystick 2 (rear connector position) |
| 23 | STR\$ (String) | STR\$ (<Expression> To convert a value variable into a character-string. | 10_A = -123 20_B = 567 30_PRINT_STR\$ (A) (For displaying -123) 40_PRINT_STR\$ (B) (For displaying _567) |
| 24 | TAB (Tabulation) | TAB (<Expression> $0 \leq \langle \text{Expression} \rangle \leq 255$ To move the cursor to the position specified by the expression for tabulating to the display (print) position. | 10_PRINT_TAB (2) ; "ABC" ; TAB (6) ; "DEF" (To display as _ _ABC_ _DEF) |
| 25 | USR (User) | USR (<Expression 1> [, <Expression 2> [, <Expression 3>]]) Value of <Expression 1>: set in the program counter. Value of <Expression 2>: set in the index register. Value of <Expression 3>: set in the B and A register. If <Expression 2> and <Expression 3> are omitted, 0 is set. To set a value in the register of the computer. A machine language program is accessed as a subroutine. The main routine is restored when the subroutine is completed. | The machine language program at address \$1000 of the memory is accessed as a subroutine. 0 is set in the index register, 1 is set in the B register, and 64 (\$40) is set in the A register. G = USR (\$1000, 0, \$0140) The value of the B and A registers when returning from the machine language program are supplied to G after execution. |

| No. | Name | Format and Function | Example and Summary |
|-----|-------------------------------------|---|---|
| 26 | VAL (Value) | VAL (<Character-string variable>) To convert the character-string into a value. | 10_ A\$ = "12" 20_ B\$ = "34" 30_ C\$ = A\$ + B\$ 40_ C = VAL(A\$) + VAL(B\$) 50_ PRINT_ C\$ (The value is 1234) 60_ PRINT_ C (The value is 46) |
| 27 | VARPTR (Variable Pointer) | VARPTR (<Variable name>) To access the start address of the first byte where a specified variable is stored in memory. | A = VARPTR(C\$) The start address in which the character-string variable C\$ is stored. A = VARPTR(D(0)) The start address in which the array variable D(0) is stored. |

8. Command of The Monitor

Press the **M O N** and **RETURN** keys to change the BASIC command mode to the Monitor mode.

Monitor commands must be capitalized.

| No. | Name | Format and Function | Example and Summary |
|-----|------------------------------|---|--|
| 1 | D command (Dump) | D [<Address>] RETURN To display the storage contents of the memory. | >_ D7000 RETURN The storage contents of 128 bytes are displayed which begin from address \$7000. |
| 2 | G command (Go) | G [<Address>] RETURN To execute the program from the specified address. | >_ G7000 RETURN The program from address \$7000 is executed. |
| 3 | M command (Modify) | M [<Address>] RETURN To check the storage contents of the memory and to change them. | >_ M7000 RETURN 7000 FF-█: waiting of data input. <Data> RETURN : To change the data. |

9. Error Messages

| No. | Error Message | Meaning |
|-----|------------------------|--|
| 1 | CONT error | The CONT command is only effective when the program is interrupted by the STOP sentence or CTRL + C command. The program cannot be continued by the CONT command after the program is interrupted and then edited, or after a hard break. |
| 2 | Division by zero error | <ul style="list-style-type: none"> • Division by 0 is performed or zero is multiplied by a power of a negative number. • Alternatively, the content of <expression> is 0 in the MOD (N, <Expression>). |
| 3 | File access error | <ul style="list-style-type: none"> • An INPUT sentence is entered for a un-opened file or a file input or output after EOF was performed. • A file already opened is opened again. |
| 4 | Illegal fn-call error | <ul style="list-style-type: none"> • A function value is out of the specified range. • Power method is applied to a negative value by a fractional number. |
| 5 | Line too long error | <ul style="list-style-type: none"> • A program line being edited is over 80 characters long. |
| 6 | Nesting error | <ul style="list-style-type: none"> • The value variable of the "FOR" and the variable of "NEXT" do not coincide. |
| 7 | Out of data error | <ul style="list-style-type: none"> • Data which is to be input by the READ command is not defined in the DATA command. • Number of input data by INPUT sentence # is not sufficient. • No DATA statement is present at a line number specified by a RESTORE sentence. |

| No. | Error Message | Meaning |
|-----|------------------------|--|
| 8 | Out of memory error | <ul style="list-style-type: none"> • The memory area for variables is insufficient. The stacked area for FOR-NEXT and GOSUB-RETURN is insufficient. • The number of output characters by a PRINT # sentence is over 255 and the memory area of the buffer is insufficient. |
| 9 | Overflow error | <ul style="list-style-type: none"> • The input value, the calculated results, or the value of a function is larger or smaller than an allowable value. • A hexadecimal constant is bigger than \$FFFF. |
| 10 | Redim'd array error | The same array name is used for two definitions. |
| 11 | RETURN error | The RETURN sentence does not correspond to a GOSUB command. |
| 12 | Subscript error | <ul style="list-style-type: none"> • An addition to an array variable is out of the range defined by a DIM sentence. • Number of additions to an undimensioned array variable is over ten. |
| 13 | Syntax error | The format is improper such as omission of a space after a function word, or the spelling is improper. |
| 14 | Tape read error | <ul style="list-style-type: none"> • Check that output data does not coincide input data. • Volume level of the cassette is improperly adjusted. |
| 15 | Type mismatch error | Character-string data is substituted for variables by an INPUT # sentence or a READ sentence. |
| 16 | Undef'd line-no. error | A line number specified by DELETE, GOSUB, GOTO, RESTORE, RUN or EOF is not found. |

| No. | Error Message | Meaning |
|-----|---------------|--|
| 17 | Value error | Parameters specified by commands or statements are out of the predetermined range. |
| 18 | Verify error | The contents of a program saved on a cassette tape is different from that in memory. |

9. SPECIFICATIONS

| | |
|---------------------|--|
| CPU: | MN 1800A (equivalent to 6802) |
| Memory: | <ul style="list-style-type: none">• 16-KB ROM• 32-KB RAM• 2-KB video RAM• 2-KB character RAM |
| Keyboard: | <ul style="list-style-type: none">• mode: sub CPU transfer• key: 63 alphanumeric keys, graphic keys and function keys |
| Display interface: | <ul style="list-style-type: none">• mode: RGB separate sync Composite video system (color/monochrome) RF modulation (Ch-3/4: Color/monochrome)• screen: 32 x 24 matrix• color: color selection for each character and background in eight colors (black, blue, red, magenta (purple), green, cyan (sky blue), yellow and white)• character configuration: alphanumeric characters (6 x 7 dots): 95 graphic symbols (8 x 8 dots): 64 other symbols (6 x 7 dots): 17 user's definition characters: 64• graphic: 64 x 48 dot matrix, color selection for each dot out of the eight colors |
| Music: | <ul style="list-style-type: none">• specification: encompasses 5 octaves and allows performance of a melody consisting of triad chords.• output: internal speaker (with volume control) and an external speaker jack |
| Cassette interface: | <ul style="list-style-type: none">• mode: FSK mode 1200 Hz (mark) and 2400 Hz (space)• baud rate: 600/2400 baud rate |
| Printer interface: | <ul style="list-style-type: none">• terminal: centronics standard applicable to printer JR-P02U |
| Joystick: | <ul style="list-style-type: none">• two terminals for joysticks |
| Program language: | <ul style="list-style-type: none">• JR-BASIC 5.0 |

Constants:

- decimal number: 10^{-39} to 10^{+38} and 0
- hexadecimal number: \$0 to \$FFFF
- character constant: 255 characters max.

Operations: addition (+), subtraction (—), multiplication (*), division (/), exponent (^), and remainder (MOD)

Command and sentences: AUTO, BEEP, CLEAR, CLOSE, CLS, COLOR, CONT, DATA DELETE, DIM, END, FIND, FOR~NEXT, GOSUB~RETURN GOTO, HCOPY, IF~THEN, INITP, INPUT, INPUT#, LET LFIND, LIST, LLIST, LOAD, LOCATE, LPRINT, MERGE MLOAD, MON, MSAVE, NEW, ON~GOSUB, ON~GOTO OPEN, PICK, PLAY, PLOT, POKE, PRINT, PRINT# RANDOMIZE, READ, REM, RESTORE, RUN, SAVE, SOUND, STOP, TEMPO, VERIFY

Functions: ABS, ASC, CHR\$, COS, EXP, FRE, HEX\$, HPOS, VPOS, INT LEFT\$, LEN, LOG, MID\$, MOD, PEEK, RIGHT\$, RND, SGN SIN, SPC, SQR, STICK, STR\$, TAB, USR, VAL, VARPTR

Monitor command: D command, G command, M command

Power source: AC 120 V \pm 10% / 230 V \pm 15%, 50/60 Hz, 10 W max.

Dimensions and weight: 56 x 348 x 208 mm (2-3/16" x 13-7/10" x 8-1/5");
1.8 kg (5-1/16 lbs) main unit

Attachments: RF cable and recording cable

10. TROUBLESHOOTING

If trouble occurs, check the following items. If trouble does not correspond any check item, contact the nearest service center.

| Symptom | Cause | Remedy |
|-----------------------------|--|---|
| No display | Power is not ON. | Check the power source and the power cord. |
| | Connections are incomplete. | Check the connections (see page 5) and firmly insert the plugs. |
| | The channel of the computer does not correspond to that of the TV set. | Check whether or not the channel selection switch (Ch-3 or Ch-4) is properly set corresponding to the channel on the TV set. |
| No color | The color selection switch is not properly set. | Check whether or not the color/monochrome selection switch at the bottom of the main unit is set to the "color" position. |
| | The color TV monitor is not properly adjusted. | Perform the color correction of the TV monitor by referring to the TV operating manual. |
| | A program is not properly executed. | Check whether or not the COLOR sentence is properly entered. |
| Nonreadable characters | The monochrome selection switch is not properly set. | Check whether or not the color/monochrome selection switch to the "monochrome" position when you use a monochrome monitor. |
| | The TV monitor is not properly adjusted. | Perform fine tuning and color correction to improve readability of the characters. The LIST command is preferably specified for optimum readability, using a black background and white characters (COLOR, 7, 0). |
| Unstable display | The TV monitor is not properly adjusted. | Properly adjust the tuning, horizontal sync and vertical sync controls. |
| Irregular character display | The computer is improperly initialized. | Turn off the power switch, wait 15 seconds, and turn it on again. |

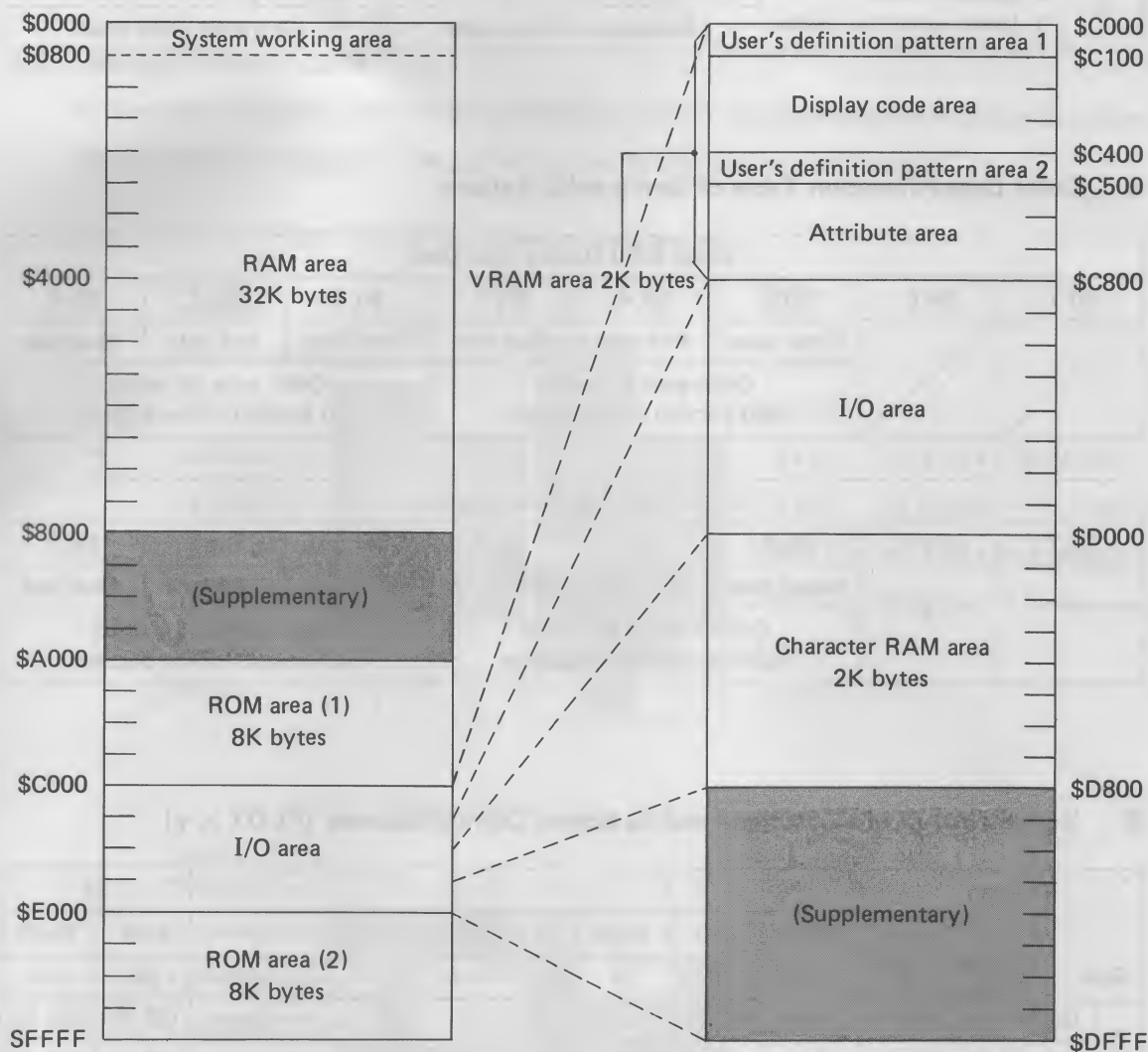
| Symptom | Cause | Remedy |
|---|---|---|
| No keyboard control | The computer over-runs (an error occurs in the program). | Press the BREAK key. If trouble is not eliminated, turn off the power switch and turn it on again. Alternatively, check the program. |
| | The BREAK key is inoperative due to irregular use of the keys. | Simultaneously press the CTRL , the SHIFT , and the : keys to initialize your computer. |
| No SAVE and LOAD operations | Trouble occurs in the tape recorder. | <p>Check whether or not the recording cable is properly connected.</p> <p>Properly operate the cassette tape recorder by referring to "Tape Recorder Operation" on page 17.</p> <p>Check whether or not the transfer speed (2400 or 600 bauds) is properly set in accordance with the type of tape recorder.</p> <p>Adjust the LOAD level (relatively high levels obtained by adjusting the volume and tone controls).</p> <p>Note that any program recorded at the transfer rate of 2400 bauds may not often be loaded if the tape recorder designed for JR-200U is not used.</p> |

Japanese characters may be displayed when you use a **PRINT CHR\$** in a program or when you perform key-in operation.

However, such display does not indicate any mechanical and electrical trouble (See display code table).

APPENDIX

1. Memory Map



2. Color Code Table

| Color | Black | Blue | Red | Magenta (Purple) | Green | Cyan (Light Blue) | Yellow | White |
|-------|-------|------|-----|------------------|-------|-------------------|--------|-------|
| Code | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

3. Attribute Area Data Allocation Table

| Video RAM Attribute Area Data | | | | | | | |
|-------------------------------|---------------------------|-----------------------|----------|-----------|----------------------|----------|-----------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | 0: Normal mode | Green data | Red data | Blue data | Green data | Red data | Blue data |
| | 1: User's definition mode | Background color code | | | Character color code | | |

4. Color Data Allocation Table of Semigraphic Pattern

| Video RAM Display Area Data | | | | | | | |
|-----------------------------|-------|---|----------|-----------|--|----------|-----------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Green data | Red data | Blue data | Green data | Red data | Blue data |
| | | Color code for upper right portion of the pattern | | | Color code for upper left portion of the pattern | | |

| Video RAM Attribute Area Data | | | | | | | |
|-------------------------------|-------|--|----------|-----------|---|----------|-----------|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1 | 0 | Green data | Red data | Blue data | Green data | Red data | Blue data |
| | | Color code of the lower right portion of the pattern | | | Color code of the lower left portion of the pattern | | |

5. Variables of PLOT Command and its Screen Correspondence (PLOT x, y)

| Column | | | 0 | | 1 | | 31 | |
|--------|-------|-------------------|---------|---------|---------|---------|----------|----------|
| | | | Left | Right | Left | Right | Left | Right |
| Row | | ^x y | 0 | 1 | 2 | 3 | 62 | 63 |
| | | | | | | | | |
| 0 | Upper | 0 | (0, 0) | (1, 0) | (2, 0) | (3, 0) | (62, 0) | (63, 0) |
| | Lower | 1 | (0, 1) | (1, 1) | (2, 1) | (3, 1) | (62, 1) | (63, 1) |
| 1 | Upper | 2 | (0, 2) | (1, 2) | (2, 2) | (3, 2) | (62, 2) | (63, 2) |
| | Lower | 3 | (0, 3) | (1, 3) | (2, 3) | (3, 3) | (62, 3) | (63, 3) |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 23 | Upper | 46 | (0, 46) | (1, 46) | (2, 46) | (3, 46) | (62, 46) | (63, 46) |
| | Lower | 47 | (0, 47) | (1, 47) | (2, 47) | (3, 47) | (62, 47) | (63, 47) |

6. User's Definition Character Function

Use a POKE sentence to directly write a user's definition character in a memory so as to freely display a figure or character. The memory addresses are assigned to characters (SP) to —. Let's create a figure of a steam locomotive using a user's definition character program.

(1) Setting of Pattern Code

A character (figure) is drawn on a character pattern design sheet in a manner as shown below so as to correspond the pattern codes to the memory addresses.

〈Front Half of SL (Steam Locomotive)〉

| User's definition character code (0) | Memory addresses | | 8 4 2 1 | | | | 8 4 2 1 | | | | Pattern code | |
|--|---------------------|---|---------|---|---|---|---------|---|---|---|-----------------|----------------------------------|
| | | | 8 | 4 | 2 | 1 | 8 | 4 | 2 | 1 | | |
| (\$C080) 0 | | | | | | | 1 | 1 | | | (\$0C) | |
| (\$C081) 1 | 1 | | 1 | 1 | | | 1 | 1 | | | (\$CC) | |
| (\$C082) 2 | 2 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$FF) | $1 + 2 + 4 + = 15$ (decimal) |
| (\$C083) 3 | 3 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$FF) | 15 (decimal) = F (hexadecimal) |
| (\$C084) 4 | 4 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$FF) | $0 + 0 + 0 + 8 = 8$ (decimal) |
| (\$C085) 5 | 5 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$FF) | 8 (decimal) = 8 (hexadecimal) |
| (\$C086) 6 | 6 | | | | | | 1 | 1 | 1 | | (\$0E) | |
| (\$C087) 7 | 7 | | | | | | | 1 | | | (\$04) | |

〈Rear Half of SL〉

| User's definition character code (1) | Memory addresses | | 8 4 2 1 | | | | 8 4 2 1 | | | | Pattern code | |
|--|---------------------|---|---------|---|---|---|---------|---|---|---|-----------------|--|
| | | | 8 | 4 | 2 | 1 | 8 | 4 | 2 | 1 | | |
| (\$C088) 0 | | 1 | 1 | 1 | 1 | | 1 | | | | (\$F8) | |
| (\$C089) 1 | 1 | | 1 | | | | 1 | | | 1 | (\$49) | |
| (\$C08A) 2 | 2 | | 1 | | | | 1 | 1 | 1 | 1 | (\$4F) | |
| (\$C08B) 3 | 3 | | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$7F) | |
| (\$C08C) 4 | 4 | | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$7F) | |
| (\$C08D) 5 | 5 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | (\$FF) | |
| (\$C08E) 6 | 6 | | 1 | 1 | 1 | | | 1 | 1 | 1 | (\$77) | |
| (\$C08F) 7 | 7 | | | 1 | | | | | 1 | | (\$22) | |

In order to determine the pattern code, 8-bit (one byte) data is divided into upper and lower 4-bit data. The upper 4-bit data is represented by a hexadecimal number, and the lower 4-bit data is also represented by a hexadecimal number.

(2) Program Creation

Write user's definition character codes (represented by 1 and 0) in a memory area using POKE sentences.

■ Program Example

| | | | | |
|-------|-------|---------|-------|--|
| 1 0 | POKE | \$C080, | \$C | Store data of the front half of the SL in a memory area corresponding to the user's definition character code 0. |
| 2 0 | POKE | \$C081, | \$C C | |
| 3 0 | POKE | \$C082, | \$F F | |
| 4 0 | POKE | \$C083, | \$F F | |
| 5 0 | POKE | \$C084, | \$F F | |
| 6 0 | POKE | \$C085, | \$F F | |
| 7 0 | POKE | \$C086, | \$E | |
| 8 0 | POKE | \$C087, | \$4 | |
| 9 0 | POKE | \$C088, | \$F 8 | Store data of the rear half of the SL in a memory area corresponding to the user's definition character code 1. |
| 1 0 0 | POKE | \$C089, | \$4 9 | |
| 1 1 0 | POKE | \$C08A, | \$4 F | |
| 1 2 0 | POKE | \$C08B, | \$7 F | |
| 1 3 0 | POKE | \$C08C, | \$7 F | |
| 1 4 0 | POKE | \$C08D, | \$F F | |
| 1 5 0 | POKE | \$C08E, | \$7 7 | |
| 1 6 0 | POKE | \$C08F, | \$2 2 | |
| 1 7 0 | COLOR | 1, 6, 1 | | 1, 6, Color designation |
| 1 8 0 | PRINT | " 1 0 " | | 1 Designation of the user's definition character mode. |
| 1 9 0 | COLOR | 7, 0, 0 | | The user's definition character is displayed on the screen. |
| | | | | 0 Cancel the user's definition character mode. |
| | | | | 7, 0, Black-and-white color designation. |

■ Execution Result

RUN



How was it?

Is an SL properly displayed on the screen?

You can create any figure or character as you please.

User's Definition Character Code Allocation Table

| Character Code | Character Allocation | Memory Address | Character Code | Character Allocation | Memory Address |
|----------------|----------------------|-----------------|----------------|----------------------|-----------------|
| 20 | (SP) | \$C000 – \$C007 | 30 | 0 | \$C080 – \$C087 |
| 21 | ! | \$C008 – \$C00F | 31 | 1 | \$C088 – \$C08F |
| 22 | ▼▼ | \$C010 – \$C017 | 32 | 2 | \$C090 – \$C097 |
| 23 | # | \$C018 – \$C01F | 33 | 3 | \$C098 – \$C09F |
| 24 | \$ | \$C020 – \$C027 | 34 | 4 | \$C0A0 – \$C0A7 |
| 25 | % | \$C028 – \$C02F | 35 | 5 | \$C0A8 – \$C0AF |
| 26 | & | \$C030 – \$C037 | 36 | 6 | \$C0B0 – \$C0B7 |
| 27 | ▼ | \$C038 – \$C03F | 37 | 7 | \$C0B8 – \$C0BF |
| 28 | (| \$C040 – \$C047 | 38 | 8 | \$C0C0 – \$C0C7 |
| 29 |) | \$C048 – \$C04F | 39 | 9 | \$C0C8 – \$C0CF |
| 2A | * | \$C050 – \$C057 | 3A | : | \$C0D0 – \$C0D7 |
| 2B | + | \$C058 – \$C05F | 3B | ; | \$C0D8 – \$C0DF |
| 2C | , | \$C060 – \$C067 | 3C | < | \$C0E0 – \$C0E7 |
| 2D | — | \$C068 – \$C06F | 3D | = | \$C0E8 – \$C0FE |
| 2E | · | \$C070 – \$C077 | 3E | > | \$C0F0 – \$C0F7 |
| 2F | / | \$C078 – \$C07F | 3F | ? | \$C0F8 – \$C0FF |

| Character Code | Character Allocation | Memory Address | Character Code | Character Allocation | Memory Address |
|----------------|----------------------|-----------------|----------------|----------------------|-----------------|
| 40 | @ | \$C400 – \$C407 | 50 | P | \$C480 – \$C487 |
| 41 | A | \$C408 – \$C40F | 51 | Q | \$C488 – \$C48F |
| 42 | B | \$C410 – \$C417 | 52 | R | \$C490 – \$C497 |
| 43 | C | \$C418 – \$C41F | 53 | S | \$C498 – \$C49F |
| 44 | D | \$C420 – \$C427 | 54 | T | \$C4A0 – \$C4A7 |
| 45 | E | \$C428 – \$C42F | 55 | U | \$C4A8 – \$C4AF |
| 46 | F | \$C430 – \$C437 | 56 | V | \$C4B0 – \$C4B7 |
| 47 | G | \$C438 – \$C43F | 57 | W | \$C4B8 – \$C4BF |
| 48 | H | \$C440 – \$C447 | 58 | X | \$C4C0 – \$C4C7 |
| 49 | I | \$C448 – \$C44F | 59 | Y | \$C4C8 – \$C4CF |
| 4A | J | \$C450 – \$C457 | 5A | Z | \$C4D0 – \$C4D7 |
| 4B | K | \$C458 – \$C45F | 5B | [| \$C4D8 – \$C4DF |
| 4C | L | \$C460 – \$C467 | 5C | ¥ | \$C4E0 – \$C4E7 |
| 4D | M | \$C468 – \$C46F | 5D |] | \$C4E8 – \$C4EF |
| 4E | N | \$C470 – \$C477 | 5E | ^ | \$C4F0 – \$C4F7 |
| 4F | O | \$C478 – \$C47F | 5F | — | \$C4F8 – \$C4FF |

7. Use of Character Generator

The character generator of JR-200U consists of a RAM, so that the user can change the storage contents, using a POKE sentence or the M command for the monitor.

In the same procedure as in the user's definition character, a character defined by the computer can be defined as a different character. For example, letter "A" may be redefined as letter "X". The redefined character can be directly entered from the keyboard or using a PRINT CHR\$ (character code) sentence so as to display it on the screen.

If you wish to change again the redefined character to have the original figure, turn off the power switch and redefine letter "X" (originally letter "A") as letter "A".

The start address of the memory area for storing any character code of the corresponding character pattern can be calculated by the following relation (the character pattern of one character is assigned to 8 data bytes)

(Relation):

$$\text{Start Address of Character Memory} = \$D000 + 8 \times \text{character code}$$

8. Display Code Table

| Upper Lower 4 bits 4 bits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|------------------------------------|---|--------|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | ~ (sp) | 0 | @ | P | ` | p | | | | | | | | ○ | □ |
| 1 | | ÷ | ! | ! | A | Q | a | q | ♠ | | | | | | ◻ | ◻ |
| 2 | | α | ▽ | 2 | B | R | b | r | ♥ | | | | | | ◻ | ◻ |
| 3 | | β | # | 3 | C | S | c | s | ♦ | | | | | | ◻ | ◻ |
| 4 | | γ | \$ | 4 | D | T | d | t | ♣ | | | | | | ◻ | ◻ |
| 5 | | Σ | % | 5 | E | U | e | u | ♠ | | | | | | ◻ | ◻ |
| 6 | | θ | & | 6 | F | V | f | v | ← | | | | | | ◻ | ◻ |
| 7 | | ▢ | ▽ | 7 | G | W | g | w | ↓ | | | | | | ◻ | ◻ |
| 8 | | ▢ | (| 8 | H | X | h | x | ↑ | | | | | | ◻ | ◻ |
| 9 | | ▢ |) | 9 | I | Y | i | y | → | | | | | | ◻ | ◻ |
| A | | ▢ | * | : | J | Z | j | z | 人 | | | | | | ◻ | ◻ |
| B | | ♪ | + | ; | K | [| k | } | ☹ | | | | | | ◻ | ◻ |
| C | | ⊙ | , | < | L | ¥ | l | | ▢ | | | | | | ◻ | ● |
| D | | ▢ | - | = | M |] | m | } | ▢ | | | | | | ◻ | ◻ |
| E | ☎ | ⊙ | · | > | N | ^ | n | - | ■ | ⊗ | | | | | ◻ | ◻ |
| F | π | | / | ? | O | - | o | | ▢ | ▢ | | | | | ◻ | ◻ |

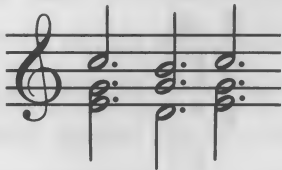
Note: Display and characters respectively indicated by shaded areas are actually assigned to Japanese characters.

9. Character Code Table (including characters except for the ASCII codes)

| Upper Lower 4 bits 4 bits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|------------------------------------|------------|--------------|------|---|---|---|---|-------|-----------|---|---|---|---|---|---|---|
| 0 | NULL | | (sp) | 0 | @ | P | ` | p | (Graphic) | | | | | | ○ | □ |
| 1 | | | ! | ! | A | Q | a | q | ♠ | | | | | | ◻ | ◻ |
| 2 | | | ▽ | 2 | B | R | b | r | ♥ | | | | | | ◻ | ◻ |
| 3 | BRE AK | INS | # | 3 | C | S | c | s | ♦ | | | | | | ◻ | ◻ |
| 4 | | (Al- pha) | \$ | 4 | D | T | d | t | ♣ | | | | | | ◻ | ◻ |
| 5 | | | % | 5 | E | U | e | u | ♠ | | | | | | ◻ | ◻ |
| 6 | | HCO PY | & | 6 | F | V | f | v | ← | | | | | | ◻ | ◻ |
| 7 | | | ▽ | 7 | G | W | g | w | ↓ | | | | | | ◻ | ◻ |
| 8 | RUB OUT | CAN SEL | (| 8 | H | X | h | x | ↑ | | | | | | ◻ | ◻ |
| 9 | | |) | 9 | I | Y | i | y | → | | | | | | ◻ | ◻ |
| A | | LINS | * | : | J | Z | j | z | 人 | | | | | | ◻ | ◻ |
| B | HOME | | + | ; | K | [| k | } | ☹ | | | | | | ◻ | ◻ |
| C | CLS | → | , | < | L | ¥ | l | | ▢ | | | | | | ◻ | ● |
| D | RET URN | ← | - | = | M |] | m | } | ▢ | | | | | | ◻ | ◻ |
| E | | ↑ | · | > | N | ^ | n | - | ■ | ⊗ | | | | | ◻ | ◻ |
| F | | ↓ | / | ? | O | - | o | (DEL) | ▢ | ▢ | | | | | ◻ | ◻ |

Note: Display and characters respectively indicated by shaded areas are actually assigned to Japanese characters.

10. Musical Performance Program



- Let's convert musical notes written on the musical scale to hexadecimal codes in accordance with the musical note lengths and the musical scale table shown on page 61.
- When you finish converting them, arrange them in the order of note lengths for higher, middle and lower tones.

Higher tones: | \$48, \$25 | \$48, \$25 | \$48, \$25

Middle tones: | \$48, \$20 | \$48, \$20 | \$48, \$20

Lower tones: | \$48, \$1D | \$48, \$1B | \$48, \$1D

How was it? We are sure that you can convert the notes as described above.

- Write data using the monitor command. When the monitor command is used, the computer is change from the BASIC control to the monitor control.

Ready

MON ■ ← [CTRL] + [M] [RETURN]

> ■ ← Upper left on the screen

Command for changing the memory contents

> M 3 0 0 0 ← Addresses at which the higher tones are written

3 0 0 0 0 0 - ■
Cursor

> M 3 0 0 0 → To access the start address

3 0 0 0 0 0 - 4 8

3 0 0 1 F F - 2 5

3 0 0 2 0 0 - 4 8

3 0 0 3 F F - 2 4

3 0 0 4 0 0 - 4 8

3 0 0 5 F F - 2 5

3 0 0 6 0 0 - 0

3 0 0 7 F F -

Enter 0 to indicate an end of a musical piece

Simultaneously press the CTRL and C keys.

> M 3 0 1 0

3 0 1 0 0 0 - 4 8

3 0 1 1 F F - 2 0

3 0 1 2 0 0 - 4 8

3 0 1 3 F F - 2 0

3 0 1 4 0 0 - 4 8

3 0 1 5 F F - 2 0

3 0 1 6 0 0 - 0

3 0 1 7 F F -

Middle tones

> M 3 0 2 0

3 0 2 0 0 0 - 4 8

3 0 2 1 F F - 1 d

3 0 2 2 0 0 - 4 8

3 0 2 3 F F - 1 b

3 0 2 4 0 0 - 4 8

3 0 2 5 F F - 1 d

3 0 2 6 0 0 - 0

3 0 2 7 F F -

Press the BREAK key located at the upper left corner of the main unit.

Ready




Use the monitor command again to display the memory contents.

> D 3 0 0 0

Command for displaying the memory contents

| | | | | | | | | |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| 3 0 0 0 | 4 8 | 2 5 | 4 8 | 2 4 | 4 8 | 2 5 | 0 0 | F F |
| 3 0 0 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 1 0 | 4 8 | 2 0 | 4 8 | 2 0 | 4 8 | 2 0 | 0 0 | F F |
| 3 0 1 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 2 0 | 4 8 | 1 D | 4 8 | 1 B | 4 8 | 1 D | 0 0 | F F |
| 3 0 2 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 3 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 3 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 4 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 4 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 5 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 5 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 6 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 6 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 7 0 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |
| 3 0 7 8 | 0 0 | F F | 0 0 | F F | 0 0 | F F | 0 0 | F F |

>  ← Press the BREAK key located at the upper right corner of the main unit.

- Press the **BREAK** key to set the BASIC command, and perform the following key input operation.

PLAY _ "F", \$3000, \$3010, \$3020 **RETURN**

The above example is very short. However, even if the program is long, the operating principles are the same as described above. Use \$00 as data for the pitch of a rest note.

- The tempo of the musical piece is specified by a TEMPO sentence. When the power switch is turned on, the initial value of TEMPO is set to be 109, which corresponds to a musical tempo 120 (quarter notes = 120). If you wish to change the musical tempo A (quarter notes = A), use the following relation $(120 \times 109)/A$.
- "F" in the PLAY sentence indicates the foreground. When the musical performance is completed, the next program is executed. When "B" is specified, the background is selected. You can execute any other BASIC program while listening to the music.

Let's play music with JR-200U.

Rest note = 0

[illegible]

| Code | 2 | | 4 | | 7 | | 9 | | B | | E | | 10 | | 13 | | 15 | | 17 | | 1A | | 1C | | 1F | | 21 | | 23 | | 26 | | 28 | | 2B | | 2D | | 2F | | 32 | | 34 | | 37 | | 39 | | 3B | |
|-----------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
| | 66 | 73 | 83 | 87 | 96 | 110 | 123 | 130 | 146 | 165 | 175 | 196 | 220 | 247 | 261 | 292 | 330 | 350 | 396 | 440 | 494 | 523 | 584 | 660 | 700 | 792 | 880 | 988 | 1046 | 1168 | 1320 | 1400 | 1584 | 1760 | 1976 | | | | | | | | | | | | | | | |
| Frequency | 66 | 73 | 83 | 87 | 96 | 110 | 123 | 130 | 146 | 165 | 175 | 196 | 220 | 247 | 261 | 292 | 330 | 350 | 396 | 440 | 494 | 523 | 584 | 660 | 700 | 792 | 880 | 988 | 1046 | 1168 | 1320 | 1400 | 1584 | 1760 | 1976 | | | | | | | | | | | | | | | |

The frequency is indicated by an approximate value.

Note length of music data

| Musical note | Symbol | Data (decimal) | Data (hexadecimal) |
|--------------------|----------------------------|-------------------|-----------------------|
| one-sixteenth note | J (7) | 6 | 6 |
| one-eighth note | J. (7) | 9 | 9 |
| one-quarter note | J. (7) | 12 | C |
| half note | J. (7) | 18 | 12 |
| whole note | J. (7) | 24 | 18 |
| | J. (7) | 36 | 24 |
| | J. (7) | 48 | 30 |
| | J. (7) | 72 | 48 |
| | J. (7) | 96 | 60 |

Data excluded from the above table can also be used.

Matsushita Electric Trading Co., Ltd.

Trade Center P.O. Box 18 Tokyo 105

YLEX7600470J
Printed in Japan